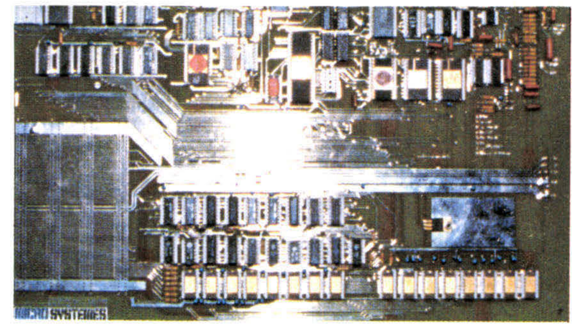




MICRO SYSTEMES

MICROPROCESSEURS / MICRO-ORDINATEURS / INFORMATIQUE APPLIQUÉE

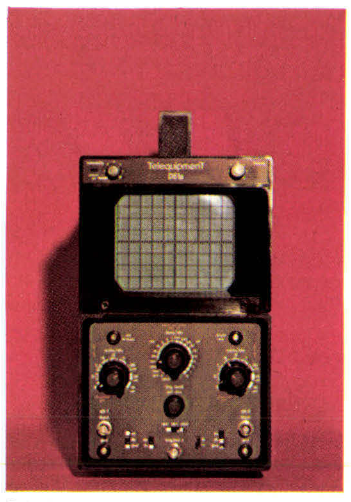
N° 3 Bimestriel – Janvier/Février 1979

10F

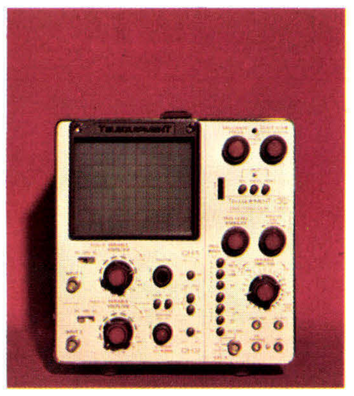
"DU MICROPROCESSEUR AU MICRO-ORDINATEUR" FORMER 1000 PERSONNES EN UN JOUR ?

Suisse 4,50 Fs - Canada 2,50 $C - Italie 2.000 Lires - Espagne 130 Pes. - Algérie 10 Din. - Tunisie 1.000 Mil.

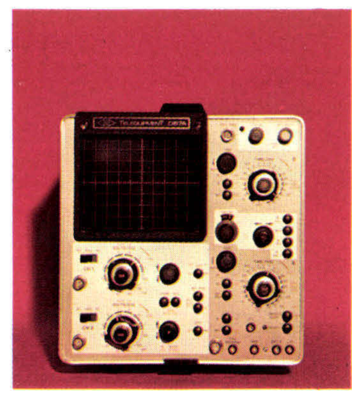
Chez PENTASONIC 6 mois de crédit gratuit sur les oscilloscopes.



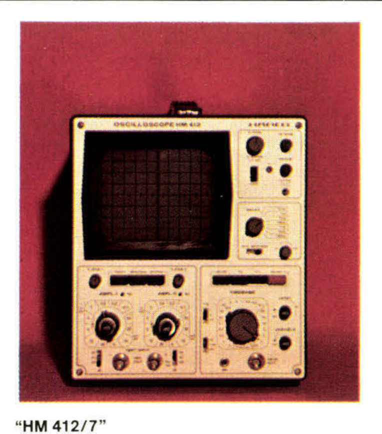
D 61 A.



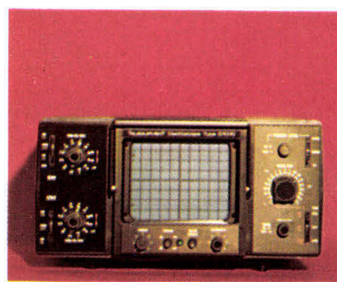
D 65.



D 67 A.



"HM 412/7"



D 1010 -

TELEQUIPMENT

1700 F — **S 61 - 5 MHz**
Dimensions 28 x 16 x 37 cm. Tube 8 x 10 cm. Grande luminosité. Ampli vertical. bande pass. : 0 à 5 MHz.

2820 F — **D 61 A. Double trace 10 MHz**
Surface utile de l'écran : 8 x 10 cm. Bande passante : 10 MHz à 10 mV/cm.

4854 F — **D 65. Double trace 15 MHz**
Surface utile de l'écran : 8 x 10 cm. Bande passante : 15 MHz à 10 mV/cm.

7680 F — **D 67 A. Double trace 2 x 25 MHz**
1 mV/cm à 50 V/cm. Double base de temps. Balayage retardé. Déclenchement : normal, AC, DC, TV ligne et trame, automatique, HF-REJ.

NOUVELLE GAMME "D 1000"

2920 F — **D 1010 - Double trace 10 MHz**
5 mV à 20 V/div. Tension maxi 500 V. Balayage 0,2 s à 0,2µS/div., 40 ns en X5.

3231 F — **D 1011 - double trace 10 MHz**
version plus performante du 1010 mais caractéristiques principales identiques.

3880 F — **D 1015 - Double trace 10 MHz**
5 mV - 20 V/div. Tension maxi 500 V - Balayage 0,2 s à 0,2µS/div. 40 ns en X5.

4464 F — **D 1016 - Double trace 15 MHz**
version plus performante du 1015 mais caractéristiques principales identiques.

Les 4 appareils de cette nouvelle série sont conçus à partir d'une technologie avancée, garantie d'une grande simplicité d'emploi. Ils sont particulièrement adaptés à l'enseignement, au dépannage TV, à la fabrication et au contrôle industriel.

HAMEG

1445 F — **"HM 307"**
Simple trace. DC - 10 MHz (– 3 dB). Entrée à 12 possibilités ± 5 %. 5 mVcc - 20 Vcc/div.

2446 F — **"HM 312" Double trace 2 x 10 MHz**
Sensibilité 5 mV/cm à 20 V/cm. Déclenchement LPS - Tube 8 x 10 cm.

3269 F — **"HM 412/7" Double trace 2 x 15 MHz**
Tube 8 x 10 cm. AMPLIFICATEUR VERTICAL. Bande passante DC : à 15 MHz (– 3 dB), à 20 MHz (– 6 dB). Sensib. : 5 mVcc - 20 Vcc/cm. Balayage retardé.

5045 F — **"HM 512/7" Nouveau double trace 2 x 40 MHz**
2 canaux DC à 40 MHz, ligne à retard. Sensib. : 5 mVcc - 20 Vcc/cm. Régl. fin 1:3. Dim. de l'écran : 8 x 10 cm. Graticule lumineux.

SONDES OSCILLOSCOPES
Commutation X1 - X10 sur la sonde. Prix...... **192 F**

LEADER

3580 F — **"LBO 508". Double trace 2 x 20 MHz.**
10 mV/cm. Soustraction de trace XY. Base de temps 0,5µs à 200 mS/cm.

VOC

4234 F — **"VOC 5". Double trace 15 MHz**
Bandes passantes : DC de 0 à 15 MHz (– 3 dB), AC de 2 à 15 MHz (– 3 dB).

CREDIT IMMEDIAT

★★★★★★★★★

L'expédition de nos appareils n'est pas gratuite, mais :

- *Ils voyagent aux risques et périls de PENTASONIC.*
- *Ils ne sont pas expédiés par la poste, ni par la S.N.C.F., mais par un transporteur.*
- *Ils sont assurés. Si jamais un de nos appareils présente à l'arrivée (vérifiez avec le transporteur) le moindre défaut d'aspect, il vous sera immédiatement changé à nos frais.*

EMBALLAGE - TRANSPORT - ASSURANCE
En contre-remboursement, **78 F** -
Avec chèque à la commande, **53 F.**

★★★★★★★★★

☎ 524-23-16
SUR LE PONT DE GRENELLE
5, rue Maurice-Bourdet - 75016 PARIS
A 50 mètres de la Maison de la Radio
Autobus : **70-72** (arrêt MAISON DE L'ORTF). MÉTRO : **Charles-Michels**

☎ 331-56-46
AUX GOBELINS
10 boulevard Arago - 75013 PARIS
Métro : Gobelins

Ouvert tous les jours (sauf dimanche) de 9 h à 12 h 30 et 14 h à 19 h 30

Sommaire

MICRO SYSTEMES





Notre couverture :

La carte imprimée du micro-ordinateur « Micro-Systèmes 1 » : 88 circuits intégrés...

Plus de 1000 personnes ont participé à notre séminaire du 25 novembre. La journée de formation la plus importante jamais organisée à ce jour (p. 78).

Président-directeur général :
Directeur de la publication :
Jean-Pierre Ventillard

Rédacteur en chef :
Alain Tailliar

Comité de rédaction :
Jean-Michel Cour — André Doris
Jean Frémaux — Jean-José Wanègue.

Secrétariat de rédaction :
Catherine Salbreux.

Rédaction :
15, rue de la Paix, 75002 Paris
Tél. : 296.46.97

Maquette : Josiane Garnier.

Publicité :
S.P.E. — **Directeur :** M. Bonvarlet.
Tél. : 296.46.97.

Abonnements :
2 à 12, rue de Bellevue
75940 Paris Cedex 19
Tél. : 200.33.05
1 an (6 numéros) : 45 F (France)
70 F (Etranger)

Société Parisienne d'Edition
Société anonyme au capital de 1 950 000 F
Siège social : 43, rue de Dunkerque, 75010 Paris
Direction - Administration - Ventes :
2 à 12, rue de Bellevue, 75940 Paris Cedex 19
Tél. : 200.33.05 - Télex PGV 230472 F

Dépôt légal 3e trimestre 78
N° éditeur : 644
Distribué par SAEM Transports Presse
Imprimerie : La Haye-Mureaux

TOUS RENSEIGNEMENTS MICROPROCESSEURS RIVE DROITE 524.23.16 RIVE GAUCHE 331.56.46 OU SUR PLACE

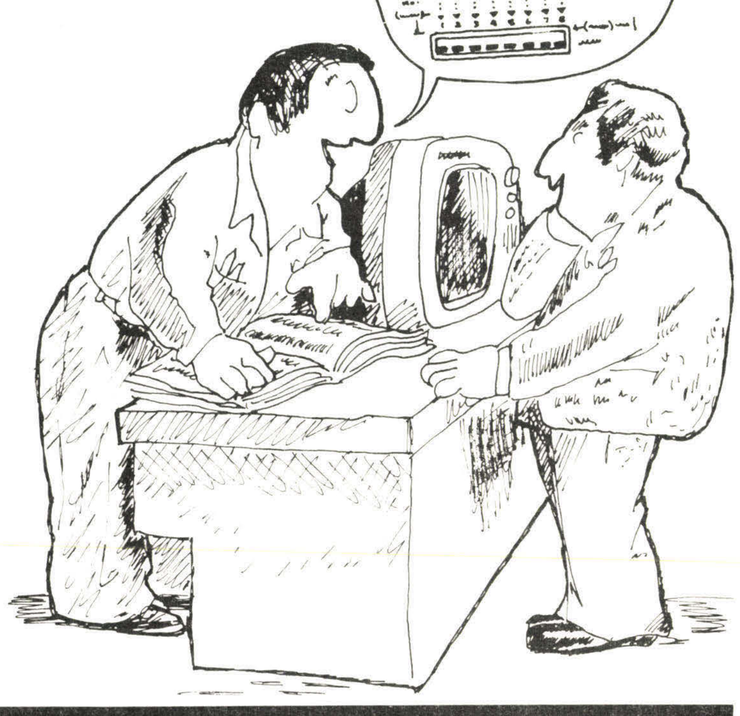
PENTA-MAGAZINE

12-78

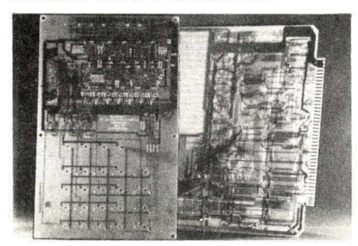
Editorial

Nous sommes anti-trust !

PENTASONIC est une association de techniciens. Nous avons ouvert un second point de vente pour le rester. Dans chaque magasin, vous trouverez une équipe légère travaillant selon ses propres méthodes, car nous sommes certains de vous apporter un meilleur service en prenant des voies concurrentes.



Le "NUMÉRO 1" ou l'anti-gadget : MKII MOTOROLA



LE fait de spécifier qu'il est à la base du 6800 MOTOROLA suffit à le considérer comme l'un des systèmes les plus souples, les plus performants, mais surtout, l'un des plus faciles à utiliser.

Il existe des microprocesseurs plus puissants mais, à notre avis, aucun de plus sympathique. Allié à sa facilité d'emploi (manuel d'utilisation en français), vous bénéficiez d'un service après-vente digne de MOTOROLA.

La vocation du MK II, outre l'initiation, est d'être le premier maillon d'un système puissant capable de gérer télétype, visu et floppy; d'être extensible grâce à ses cartes interfaces... qui existent!

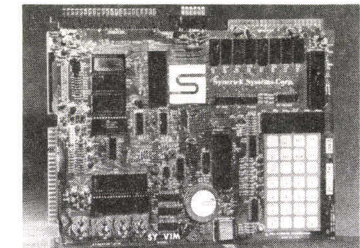
Le MK II c'est l'anti-gadget des systèmes de développement.

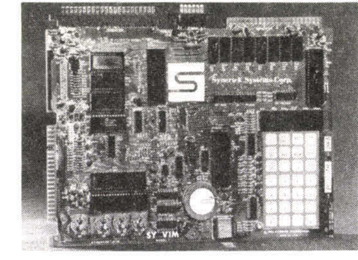
1968 F. Avec notice d'utilisation en français.

L'OUTSIDER : le VIM I SYNERTEK

Nous considérons le 6502 comme un microprocesseur très légèrement moins puissant que le 6800, bien que très proche de celui-ci. Le système de développement qui lui est associé est, par contre, très étoffé. Ses 4 K de moniteur d'origine, son 1 K de RAM, sa gestion directe de télétype ou de visu, en font un système complet. La plupart des extensions peuvent se monter sur la carte et particulièrement 3 K de RAM supplémentaires, ainsi que 32 K de ROM. Parmi ces 32 K de ROM on pourra implanter un gros BASIC de 8 K, dont Synertek annonce la commercialisation. Autre avantage le VIM I se vend tout monté.

2350 F avec notice d'utilisation en français.



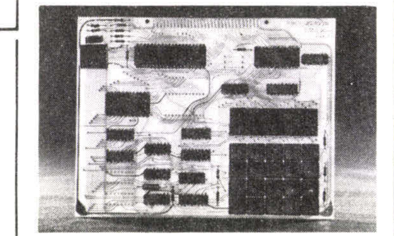


FAITES-LE DONC VOUS-MÊME!

Voulez-vous construire votre système 6800 vous-même? Nous tenons, dans ce cas, à votre disposition un classeur comprenant le circuit imprimé, les plans, et les notices d'un système de développement. Une fois monté, il gérera une visu ou un télétype. Il disposera d'1 K de mémoire, un PIA, un ACIA et un moniteur MIK-BUG. Ce kit, destiné aux amateurs avertis, bénéficie de l'assistance technique PENTASONIC. MOTOROLA le commercialise, dans sa version d'origine sous l'appellation MKI.

Le circuit imprimé 200 F.

Un petit malin qui cache son jeu



Ce produit à base de SCMP détient certainement l'un des meilleurs rapports prix/performances du marché. Particulièrement destiné aux contrôles de processus, c'est le système d'initiation par excellence. Toutes les cartes d'extension, ainsi que le produit lui-même sont fabriqués en France.

Série UC 1000 EMR, 985 F

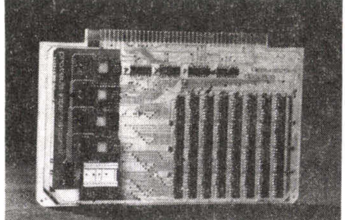
LA MEILLEURE ET LA PIRE DES CHOSES

Le soft est la meilleure et la pire des choses. La meilleure lorsqu'il s'agit d'aider l'utilisateur par une plus grande souplesse de langage, une plus grande puissance. La pire lorsqu'on tombe dans la facilité et qu'on l'utilise mal : par exemple gérer un clavier ou une visu directement. Le soft n'est pas le Bon Dieu! Il ne peut pas tout faire et s'il s'occupe à servir ses propres intérêts, il ne lui restera que peu de temps à vous consacrer.

Le but, notre but, n'est pas d'afficher des petits caractères sur un printer ou sur une télévision, mais, de travailler en concordance avec notre système. N'oubliez jamais que le soft n'est que le prolongement de vous-même : avec vos qualités et ses défauts. Du soft... PENTABUG ! qui se présente sous la forme de 1 MM 2716 se montant directement sur les emplacements EPROM du MK II et permet de brancher directement nos cartes de visualisation et nos claviers sur les KITS MAZEL et MK II • Mis au point par M. DAUTREVAUX..:.................... 195 F TTC

Comprenant 1 MM 2716 + 1 notice + 1 listing

DO YOU SPEAK BASIC?



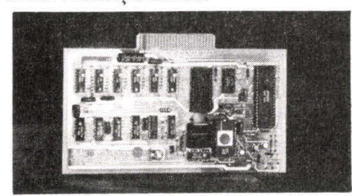
Cette carte, fabriquée par PROTEUS INTERNATIONAL vous permet de compléter votre système informatique et d'accéder à un "personal computer" de haut de gamme. Il existe différents types de Basic. Le Tiny-Basic (de 2 à 4 K), qui se rapproche plus du gadget que de l'outil de travail et qui permet entre autre d'apprendre le langage. Le Full-Basic (de 4 à 7 K) qui est le langage d'origine et l'Extended-Basic que l'on considère comme le plus souple, et l'un des plus puissants. Grâce à celui-ci, vous aurez accès à la fantastique bibliothèque existante : calcul scientifique, gestion de stock, aterrissage de fusée, jeux, etc.

Carte basic 1820 F

- Ce moniteur comprend et exécute 28 fonctions.
- Il permet en outre d'adresser un ACIA en 8010-8011 et de commander tout interface travaillant en ASC II.

DEUX SYSTÈMES DE VISUALISATION D'INFORMATIONS SUR TERMINAL VIDÉO

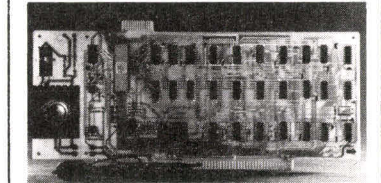
Une carte pour voir, à la française...



Cette carte est un outil de travail professionnel. L'entrée des informations se fait en ASCII, en série ou en parallèle. La mémoire de l'écran se trouve sur la carte, ainsi que la gestion de celui-ci. La sortie est en vidéo directe, c'est-à-dire qu'elle se branche directement sur l'entrée vidéo de votre téléviseur, ou, sur l'entrée UHF par l'intermédiaire d'un modulateur. Elle gère 16 lignes de 64 caractères en vitesse réglable jusqu'à 12.000 bauds en série.

Carte interface vidéo SFF 96364, 1512 F en kit, avec notice en français.

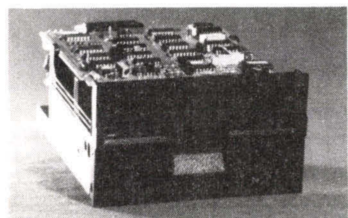
Une carte pour voir à l'américaine...



Cette carte a été conçue par MOSTEK aux U.S. et modifiée pour fonctionner sur le standard français. Elle est commercialisée montée et testée. Elle possède un avantage évident pour les radio-amateurs : elle fonctionne en Baudot à 47,5 bauds et peut faire la conversion ASCII/Baudot. L'entrée des informations se pratique en série ou en parallèle, de plus, le curseur est adressable en absolu ou en relatif ce qui permet un pseudo-graphisme.

Carte interface vidéo MK 3870, 1580 F montée.

MINI FLOPPY DISK-DRIVE ou une mémoire d'éléphant



Dernier maillon de la chaîne 6800 MKII : l'extension "FLOPPY"! De la même façon que ses grands frères ordinateurs, le 6800 peut avoir besoin d'une mémoire de masse, non volatile et d'accès rapide. Cette nouvelle technologie, le Floppy, entre dans l'univers du Hobbyste.

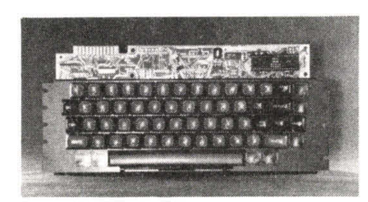
Ce lecteur offre la possibilité de stocker 2,6 mégabits de mémoire sur un disque souple avec un temps d'accès moyen aux information de 500 mS. Il peut être livré indifféremment de marque SHUGART ou PERTEC. Il s'adapte également sur APPLE. La carte de formatage pour 6800 sera disponible fin novembre et compatible avec tous les systèmes 6800.

Le lecteur double densité 2100 F, le Floppy Disk 49 F.



Carte
BUS EXORCISER
vendue par le club 6800
149 f sans connecteur

Quand la force de frappe devient caresse...



Nous vous présentons un clavier aussi moderne que votre microprocesseur. Malgré la technique employée (pas de contacts) et la sophistication de sa gestion, il est vendu à un prix très compétitif. Il devient très vite lassant, en cours de programmation de taper sur un clavier qui rebondit ou dont les touches manquent de souplesse. Sa sortie s'effectue sur 7 bits + strobe et il s'alimente uniquement en 5 volts. Il est géré par un microprocesseur GI spécialisé pour ce travail.

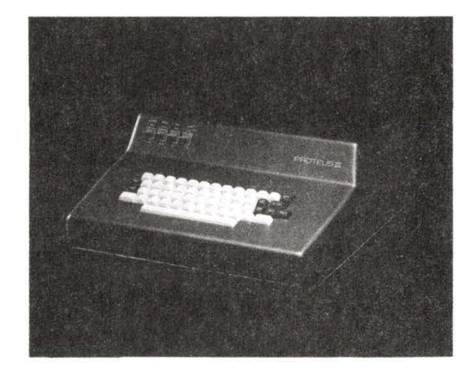
Clavier ASCII, dernier-né de KEYTRONIC® 1180 F.

IL SAVAIT TOUTES LES CHOSES ET PARLAIT TOUTES LES LANGUES...

PROTEUS : dieu grec, qu'on appelait le sage d'Egypte, car il vivait dans une petite île au large des côtes d'Alexandrie.

Proteus savait toutes les choses et parlait toutes les langues. A ce titre, il était très sollicité, les gens se pressaient pour l'interroger. Mais comme Proteus savait, en outre, prendre toutes les formes, lorsqu'il était excédé, il devenait arbre, poisson ou fleur.

Un jeune homme réussit cependant, à force de gentillesse et de patience, à séduire Proteus, qui lui livra tous ses secrets...



GÉRER POUR RÉUSSIR.

Proteus III, système complet de micro informatique de gestion, est maintenant là pour vous seconder fidèlement en traitant rapidement, efficacement, clairement tous les problèmes de classement, de comparaison, recherche, identification, que vous étiez seul jusqu'ici à pouvoir résoudre pour gérer votre activité.

Or, que vous soyez Chef d'Entreprise, Expert comptable, Médecin, Dentiste, Pharmacien, Avocat, Gérant d'immeubles, Hôtelier, Agent de voyages, Ingénieur ou Architecte, ... vous avez tous un point commun : vous savez que pour réussir, il faut gérer mieux.

PROTEUS III UNITÉ CENTRALE

25 K de mémoire ou 41 K.
16 K-Ram (Réf. Proteus III-A).
32 K-Ram (Réf. Proteus III-B).
1 K-Mémoire de page écran.
8 K-Rom (Basic).
Unité centrale 6800 travaillant à 0,894 MHz.
Clavier, 53 touches capacitives.
128 caractères dont ASCII majuscules + caractères graphiques.
Sortie UHF et vidéo.
Interface imprimante série : 20 mA, TTY, RS 232, gérée par le Basic.
Basic type étendu, capable de gérer 3 périphériques simultanément (clavier-écran vidéo, imprimante TTY, magnéto K7), plus, éventuellement 1 à 3 floppy disk.
Magnétophone à K7 travaillant à 300 Bauds, utilisé comme mémoire de masse.

Prix .... 10.758 F

PROTEUS MONITOR

Moniteur vidéo de 30 cm, haute résolution, avec écran standard Proteus (européen).

Prix .... 1.680 F

PROTEUS PRINT

Imprimante sur papier normal (non métallisé), travaille sur 80 colonnes à 1.200 Bauds (120 ch./s).

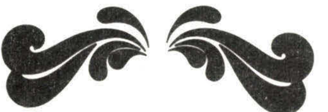
Prix .... 10.240 F

PROTEUS FLOPPY

(livraison fin novembre).
Mini-Floppy travaillant en double densité, double face.
2,6 Megabits par disquette.
Vitesse de transfert de 250 K-Bauds.
Temps d'accès moyen à un fichier : 500 ms.

Prix .... 4.704 F

PENTASONIC



☎ 524-23-16
SUR LE PONT DE GRENELLE
5, rue Maurice-Bourdet - 75016 PARIS
A 50 mètres de la Maison de la Radio
Autobus : **70-72** (arrêt MAISON DE L'ORTF), MÉTRO : **Charles-Michels**

☎ 331-56-46
AUX GOBELINS
10 boulevard Arago - 75013 PARIS
Métro : Gobelins

Editorial

Il est superflu de dire que l'usage du microprocesseur et du micro-ordinateur se développe rapidement en France. Si chacun devient quotidiennement témoin de cet irréversible raz de marée technologique, force est de reconnaître que nous n'en sommes qu'au début et que cela ne va pas et n'ira pas sans poser quelques problèmes, en particulier celui de la formation.

En supposant que l'Education Nationale puisse rapidement et efficacement adapter ses programmes pour offrir à cette nouvelle génération de techniciens et ingénieurs les connaissances indispensables à la maîtrise de cette micro-informatique, cette solution ne portera ses fruits au plus tôt que dans deux ans.

En attendant que peuvent faire les entreprises ? Certes, de nombreux ingénieurs nouvellement sortis de leurs écoles apportent avec eux les compétences requises. De plus, il y a ceux qui ont déjà pu, grâce à la formation permanente, s'initier à cette technologie au long des différents séminaires. Est-ce suffisant ? Certainement pas !

Mais connaît-on exactement les besoins réels en matière de formation sur les microprocesseurs ? Les personnes exerçant une profession dans l'électronique ou toute autre activité, pourraient-elles une fois formées à cette technologie et grâce à elle, résoudre leurs problèmes ? Pour le moment nous n'en avons aucune idée. Ajoutons à cela que tous ceux qui à titre personnel, ont de bonnes raisons de vouloir parfaire leurs connaissances dans ce domaine nouveau méritent malgré tout quelque attention.

Si nous nous intéressions à une telle étude, nous découvririons que, d'une part, l'ensemble de ces besoins dépasse largement les budgets de formation alloués dans le cadre de la Taxe de Formation Professionnelle et d'autre part qu'il n'y a pas assez de séminaires et d'animateurs pour répondre à cette demande.

Il ne nous appartient pas de dire dans ce propos, qui des Pouvoirs Publics ou de l'entreprise doit prendre en charge les frais d'une pareille opération, de même que notre vocation n'est pas de remettre en cause les prix proposés pour les séminaires.

Par contre, nous pouvons témoigner d'une expérience et ceci d'autant plus facilement que nous en étions les organisateurs. Il est aujourd'hui possible de former simultanément 1 000 personnes en un jour, et ce de façon efficace.

Pourquoi cela, et comment est-ce possible ? Il est facile de constater qu'un vaste public semble prêt à sacrifier un samedi entier pour venir suivre un séminaire pour peu que le prix en soit abordable.

Pour qu'une telle opération ne vous conduise pas à la faillite, il faut que d'emblée vous soyez certain de toucher plusieurs centaines de participants. Or ce qui très vite fait monter le coût d'un séminaire, ce sont avant tout les frais de publicité. Plus vous voulez toucher de monde et plus il devient difficile de maintenir un prix d'appel.

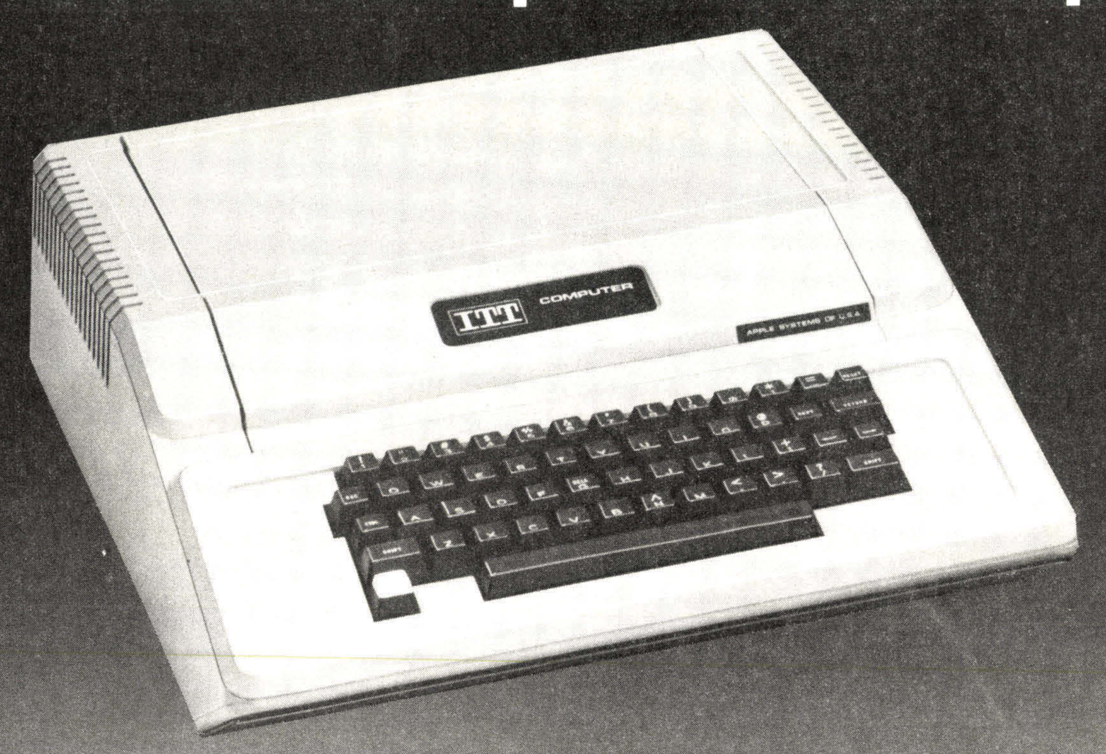
La solution à ce problème : inviter une revue, de grande diffusion, qui fait autorité en la matière à apporter son concours. C'est ainsi que MICRO-SYSTÈMES a pris en charge l'organisation de cette grande campagne de publicité et d'information propre à promouvoir et assurer le succès d'une manifestation de cette envergure.

Est-ce qu'une telle expérience sera renouvelée ? Nous ne le savons pas encore.

Ce dont nous sommes sûrs à présent, c'est que les chances de réussite existent et que face à la demande grandissante de formation, cette solution mérite d'être mentionnée auprès des Pouvoirs Publics, vers qui les entreprises n'oublieront pas de se tourner le moment venu. ■

Jean-José WANÈGUE

Le micro-ordinateur ITT
un nouveau concept de l'informatique



Les entreprises peuvent accéder aujourd'hui à une informatique sur mesure.

Le micro-ordinateur ITT est un système complet, parfaitement autonome, d'un encombrement réduit, extrêmement fiable, et dont les possibilités sont fantastiques.

D'un prix accessible, d'une grande facilité d'emploi, sa technologie avancée alliée à une grande souplesse d'utilisation lui permet d'effectuer la majorité des travaux nécessaires aux commerces, industries, aussi bien que des applications scientifiques ou personnelles.

SPÉCIFICATIONS TECHNIQUES

C'est un micro-ordinateur très complet et parfaitement autonome. Il comprend un BASIC et un MONITEUR en ROM (mémoire de 8K octets), de 4 à 48K octets de mémoire vive (RAM) graphisme couleurs, l'interface magnétophone, connecteur E/S, un clavier ASCII du type machine à écrire, une alimentation à découpage, une interface SECAM, le tout présenté dans un boîtier moulé de haute résistance.

Microprocesseur : type 6502 avec horloge interne de 1 MHz.

Visualisation vidéo : en sortie vidéo, le contenu de la mémoire est affiché soit sous forme de texte, graphique-couleurs, ou graphique à haute résolution (accès mémoire totalement transparente) dont l'affichage nécessite 8K octets de mémoire (d'où nécessité d'un minimum de 12K octets de mémoire RAM).

Texte : 24 lignes de 40 caractères (matrice de 5 x 7 points). Caractères normaux, inverses (noir sur blanc), ou clignotants. Puissant logiciel de commande en ROM.

Mémoire : elle est composée en 3 parties qui peuvent contenir des RAM de 4 à 16K octets chacune. Ces RAM additionnelles peuvent être aisément fixées sur la carte de base (jusqu'à 48 K octets) ne nécessitant donc aucune adjonction de carte supplémentaire de mémoire périphérique. Le circuit de base comporte 8K octets de ROM (mémoire permanente) contenant le BASIC et le MONITEUR.

Deux emplacements sont d'ores et déjà prévus pour l'évolution future du logiciel de base.

Entrée/sortie (E/S) : le micro-ordinateur ITT en plus du clavier ASCII comporte un modulateur RF, un interface rapide pour magnétophone (1 500 BPS), un haut-parleur, un connecteur E/S et 8 connecteurs pour interface de périphériques.

Basic : le langage spécifique au micro-ordinateur ITT est contenu dans 6K octets de ROM (permanent).

Moniteur : Mémoire permanente utilisant 2K octets ROM.

Mini-assembleur et désassembleur, etc...

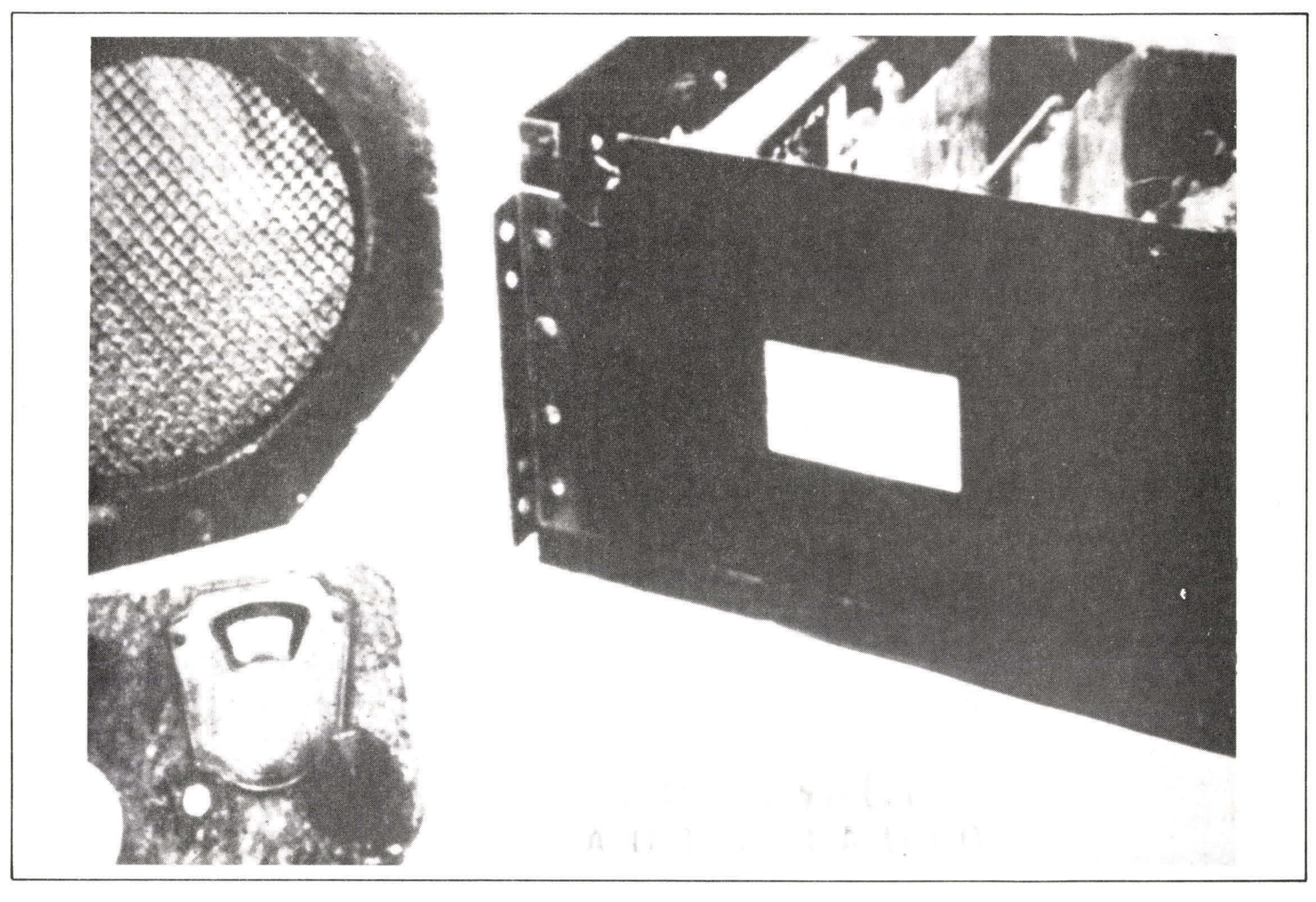


Documentation sur demande

ITT OCEANIC
DIVISION TÉLÉ ORDINATEURS
97 AVENUE DE VERDUN
93230 ROMAINVILLE

153-09-78-ITT

Introduction des microprocesseurs dans l'électronique automobile



Une des premières applications de l'électronique à l'automobile : l'auto-radio construit par la firme Motorola à ses débuts. (Doc. Motorola.)

Pris entre les exigences contradictoires des consommateurs (sécurité, confort, vitesse) et la législation gouvernementale, (consommation d'énergie et pollution) les constructeurs d'automobiles recherchent actuellement l'utilisation des microprocesseurs comme éléments de contrôle pour répondre à ces demandes apparemment inconciliables.

L'application des semi-conducteurs à l'automobile n'est pas très nouvelle. Elle a commencé il y a plus de 20 ans avec les transistors et les auto-radios puis, ensuite avec des composants tels que les redresseurs, l'allumage électronique, les circuits de contrôle automatique d'éclairage et les circuits de puissance des essuie-glaces. Néanmoins le taux de pénétration de ces semi-conducteurs n'était pas ce qu'on peut appeler explosif. Aujourd'hui le micro-processeur permet d'envisager un changement radical de cette situation.

Facteurs à l'origine du changement

De nombreux facteurs et besoins agissent sur la conception automobile, ils peuvent être classés en cinq catégories.

Pour satisfaire ces besoins, un grand nombre d'applications électroniques se sont développées pour l'automobile et parmi elles le microprocesseur est apparu comme le composant principal.

Pratiquement, pour fonctionner, un micro-ordinateur a besoin d'une mémoire, de circuits d'entrées-sorties vers le monde extérieur et doit être capable d'exécuter un jeu d'instructions. La **figure 1** illustre les exigences de base d'un système de contrôle de moteur automobile et décrit brièvement le schéma de décisions d'un ensemble électronique. Ce schéma de décisions montre la facilité avec laquelle un micro-ordinateur peut déterminer les conditions de fonctionnement du moteur. Le microordinateur surveille les trois signaux d'entrées suivants :

— le démarreur,
— la vitesse de rotation du moteur,
— un signal analogique représentant la position de la pédale d'accélérateur.

Il peut alors adapter les règles à suivre pour les six situations qui contrôlent le fonctionnement de la voiture : point mort, starter ou manivelle, décélération, roue libre, allure moyenne et allure rapide.

Ainsi le micro-ordinateur a permis de concevoir, une série d'options qui n'étaient pas envisageables auparavant, lorsque la plupart des systèmes électroniques étaient des boîtes noires indépendantes qui ressemblaient beaucoup aux fonctions mécaniques qu'elles

Parce que les micro-ordinateurs destinés à l'automobile sont récents, aucun circuit LSI standard ne s'est encore distingué.

remplaçaient. De fait en changeant des éléments de programme de la mémoire ROM. (read only memory) d'un micro-ordinateur, la personnalité de celui-ci peut être entièrement modifiée et ainsi répondre à des exigences fonctionnelles différentes.

Les défis

Plusieurs défis doivent être relevés pour qu'un micro-ordinateur puisse trouver son rôle propre dans une voiture. Il faut d'abord faire baisser le coût des LSI/micro-ordinateurs et microprocesseurs aussi vite que possible pour atteindre le très bas niveau considéré comme nécessaire par les constructeurs automobiles. Le second défi est l'impossibilité de se procurer à des prix peu élevés des capteurs, des transducteurs, et des interfaces efficaces utilisés entre le micro-ordinateur et les variables du monde réel de la voiture (vitesse, température, pression). Si la technologie des capteurs et des transducteurs ne se développe pas rapidement dans les prochaines années, la cadence d'utilisation des micro-ordinateurs dans l'automobile sera ralentie notablement.

Personne ne sait comment aborder le problème des capteurs : on sait seulement qu'ils sont nécessaires. Ce défi sera relevé par l'industrie des semi-conducteurs, car bon nombre d'entre-eux sont une extension naturelle de cette technologie, de la même façon que le L.S.I. est en fait dérivé de la technologie des éléments discrets.

Le choix d'un micro-ordinateur

Parce que les micro-ordinateurs destinés à l'automobile sont relativement récents, et les applications

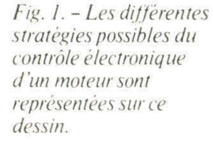


Fig. 1. – Les différentes stratégies possibles du contrôle électronique d'un moteur sont représentées sur ce dessin.

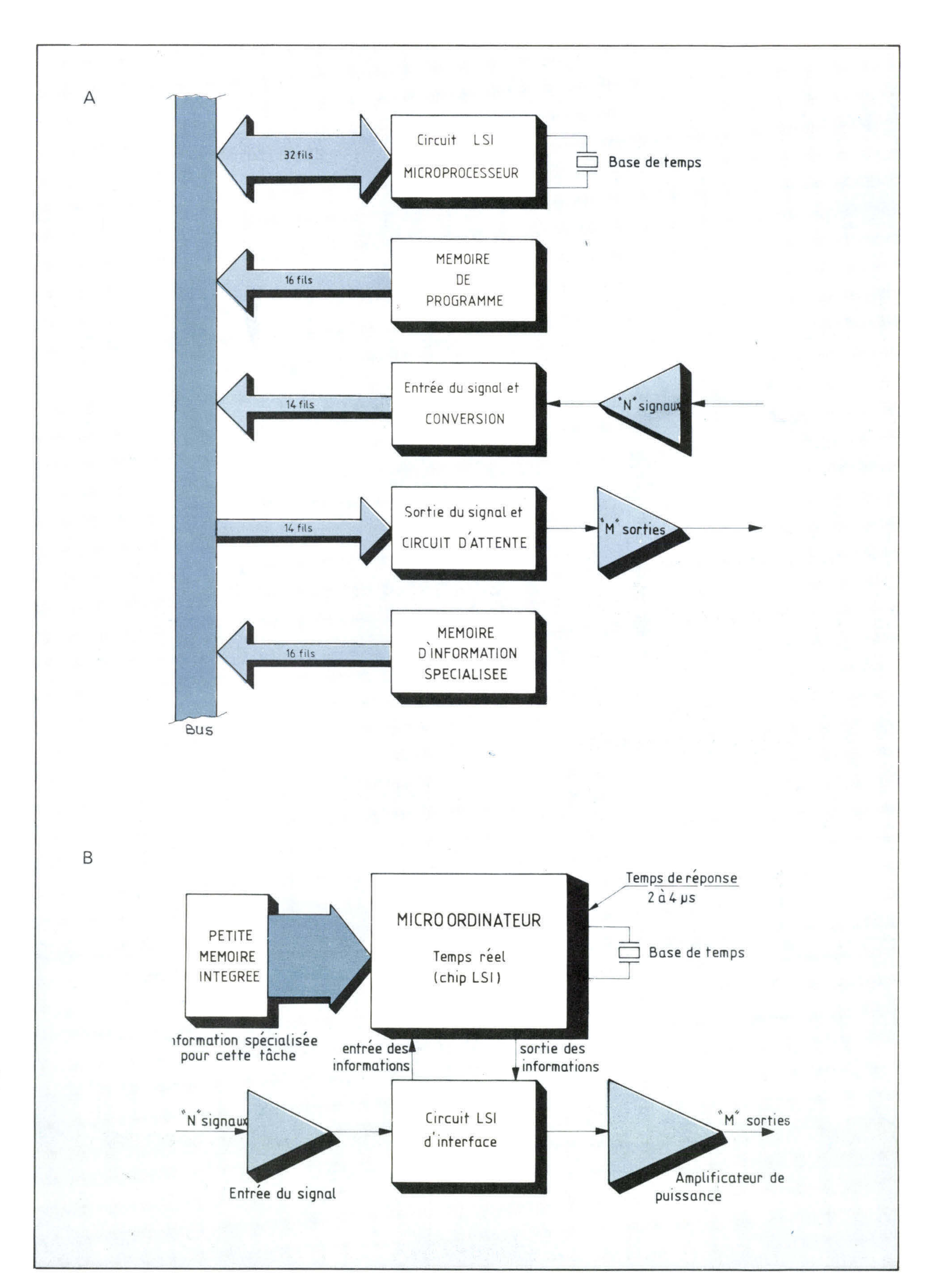


Fig. 2. – Deux approches possibles de la gestion d'un moteur d'automobile : un micro-ordinateur universel organisé autour d'un BUS (A) permet d'adapter le système à un grand nombre de fonctions ; un micro-ordinateur spécifique (B) réalise l'approche optimum d'un cas particulier.

L'évolution vers un traitement distribué permettra de réduire le nombre de câbles et d'augmenter la fiabilité du système.

nombreuses, aucun micro-ordinateur standard ne s'est encore distingué. On ne sait pas encore si une unité à vocation spécialisée ou universelle est préférable. Il y a beaucoup d'applications automobiles pour lesquelles l'utilisation d'un micro-ordinateur universel s'approche d'une solution optimale du problème de conception et les dépenses supplémentaires affectées à un micro-ordinateur spécialisé seraient évitées.

D'autre part, pour le contrôle moteur, le micro-ordinateur doit réaliser des programmes de multiplication et de division et dans ce cas un micro-ordinateur « tous usages » ne serait pas rentable.

Les plus importantes firmes automobiles sont maintenant à différents stades dans le cycle de conceptions du micro-ordinateur. Les uns vont utiliser des unités universelles comme approche principale du contrôle moteur, les autres utiliseront des unités spécialisées. Le concept d'un micro-ordinateur à large gamme d'applications dominera la production initiale pour le marché de l'automobile pendant plusieurs années. Finalement les micro-ordinateurs spécialisés prendront la tête. L'évolution vers ce type de micro-ordinateurs, sera progressive pendant 4 à 5 années avec des changements dans la mémoire ROM et des modifications occasionnelles d'interfaces lorsque des capteurs nouveaux et plus performant seront disponibles. La **figure 2** compare un micro-ordinateur universel à un micro-ordinateur spécialisé optimisé.

Architecture du système

Dans la première phase de l'application, des micro-ordinateurs à l'automobile, on utilisera un ou deux processeurs centraux, ce qui demandera une grande quantité de câbles de liaisons. Cette phase initiale sera caractérisée par une unité centrale de traitement reliée à des capteurs répartis dans le véhicule. Plus tard, la distribution des fonctions électriques dans la voiture pour des raisons de sécurité et de performances, pourrait engendrer une tendance vers des processeurs distribués. Un petit nombre de micro-ordinateurs spécialisés et de satellites semi-autonomes seraient placés dans diverses parties de la voiture. Ces unités recevraient leurs instructions d'un cerveau centralisé mais seraient pourvues d'une autonomie suffisante pour fonctionner sans l'assistance d'un processeur central. Ceci diminuerait le volume de câbles nécessaires, augmenterait la fiabilité du système et permettrait au véhicule de fonctionner même en cas de panne du processeur central.

L'évolution vers un traitement distribué permettra aux autres équipements électroniques tels que les photo-couleurs et les fibres optiques de prendre une importance considérable. Ces équipements produiront vraisemblablement des liens de communications entre processeurs centraux et satellites lorsque la quantité d'informations échangées sera très élevée menaçant ainsi le rapport signal/bruit.

Concepts de contrôle moteur

Les besoins en matériel électronique nécessaire au contrôle du moteur sont diversifiés d'où la nécessité d'adapter les exigences de performance. La variété des modèles et les variations de l'équipement dans une année de production particulière accentuent encore ce besoin. Les industries automobiles et de l'électronique ont exploré une large gamme de solutions requérant des microprocesseurs.

Des solutions d'ingéniérie systématique sont difficiles à envisager à cause des exigences fluctuantes, et il est malaisé de déterminer quels équipements complexes peuvent être produits à une cadence de plus de 100 000 unités par mois. Pour des questions de rentabilité, chaque câble, chaque capteur, chaque interface et chaque composant électrique doit fournir une contribution fonctionnelle, être fiable et avoir la plus grande simplicité physique possible. L'approche donnée ici en **figure 3** est un exemple de contrôleur qui pourrait bientôt être réalisable.

Les tâches à accomplir par ce contrôleur électronique comprennent généralement l'allumage, le contrôle du carburant, les réglages du carburateur et le recyclage d'une partie des gaz d'échappement vers celui-ci. Environ huit signaux d'entrées issus des périphériques rapportent l'état du moteur par une tension, une fréquence, des impulsions de largeur variable, ou des réseaux commutables. Il faut alors un nombre identique de sorties pour manipuler les dispositifs électromécaniques chargés d'appliquer les consignes données par le micro-ordinateur.

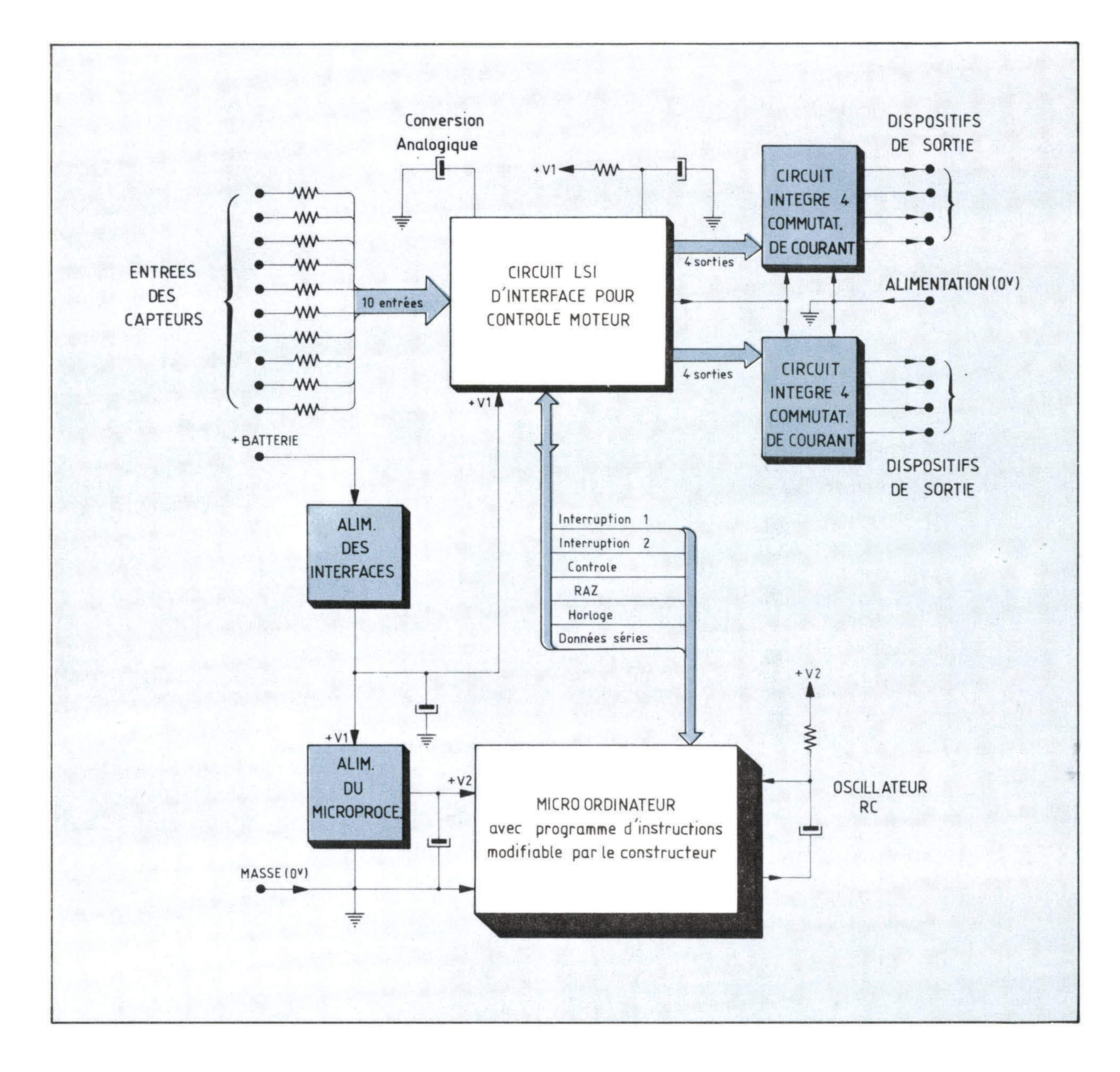
Les lignes de signaux périphériques sont généralement incorporées dans le réseau de câbles de l'automobile où y voisinent des signaux de grande amplitude et de haute fréquence. Ces lignes sont donc sujettes à des parasites ou même à des courts-circuits provoqués par des fils de bougies défectueux ou à nu. La situation se compliquera encore lorsque des changements dans les caractéristiques des signaux périphériques se produiront au fur et à mesure de la création de modèles plus rentables de capteurs.

Les signaux sont traités ou analysés par un chip interface qui adapte et transforme les informations pour le micro-ordinateur.

La **figure 3** montre dans un contexte général comment les signaux du monde extérieur sont pris en compte pour réaliser le contrôle du moteur. Le circuit interface périphérique contient la plupart des circuits de conditionnement des signaux.

Généralement, il faut gérer les sorties à l'intérieur de l'interface pour s'assurer que le micro-ordinateur ne crée pas de commandes erronées. En outre, il est nécessaire de placer des capacités d'intégration adaptées au temps de réponse du micro-processeur.

Fig. 3. – Système de contrôle total de moteur.



Dialogue entre circuits

Lorsque les connexions indispensables issues du monde extérieur et celles des services de secours sont en place, il ne reste que peu de broches disponibles pour dialoguer avec le microprocesseur. Dès lors, il est préférable de traiter quelques-uns des signaux les moins importants directement dans le microprocesseur. Une solution moins onéreuse consiste à prétraiter le signal d'entrée à l'intérieur même du circuit d'interface.

Le dialogue doit comprendre essentiellement les données suivantes :
— redémarrage du micro-ordinateur,
— sélection et transfert des signaux d'entrée,
— sélection et transfert des signaux de sortie,
— dialogue avec des systèmes de test-usines,
— chargement du programme d'instruction.

De plus, il faut utiliser plusieurs interruptions destinées à faire face à des événements simultanés dont le « timing » est très précis. On peut économiser de l'espace-mémoire en envoyant des mots de données, en série et à grande vitesse (par exemple à une fréquence de 0,5 à 2 MHz) ce qui diminue les connexions entre les interfaces et les périphériques.

L'unité centrale

Dans l'unité centrale du micro-ordinateur ou C.P.U.*, les exigences de « timing » nécessaires au traitement des résultats du contrôle-moteur tolèrent en général une variation de 5 %. En conséquence, un simple oscillateur R.C. avec compensation de température

* *CPU : Central Processing Unit.*

La fourchette de tension d'alimentation d'un micro-ordinateur pour automobile sera de 3,5 à 5,25 V.

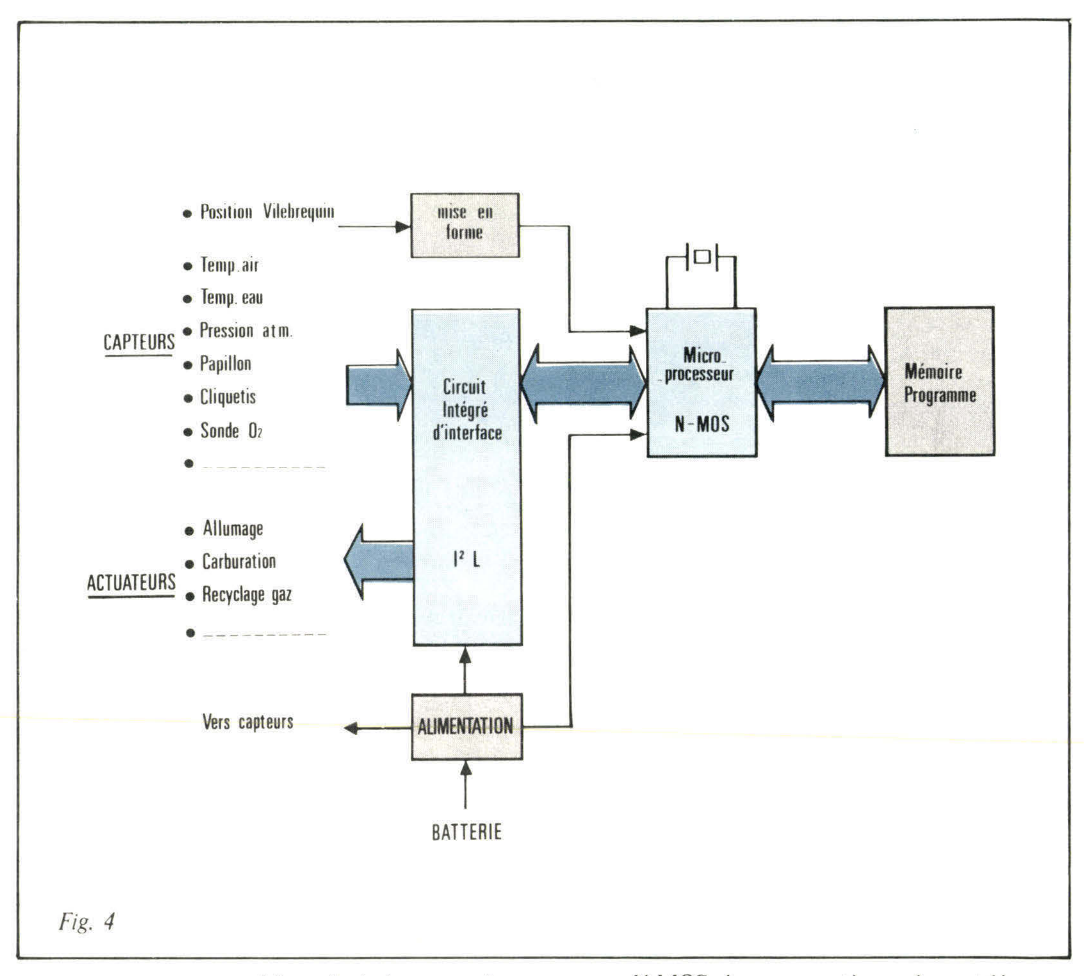


Fig. 4

Motorola intègre un microprocesseur N.MOS dans ses systèmes de contrôle pour moteur. Notez la présence d'une mémoire programmable interchangeable et par là même la grande souplesse du micro-ordinateur.

peut réaliser le signal d'horloge destiné à générer la séquence des cycles d'instructions. Des temps de cycles de 0,5 à 1,5 μs semblent être bien adaptés au fonctionnement d'un tel système.

Pendant le démarrage du moteur la batterie peut descendre jusqu'à 4,5 volts et le starter être à peine utilisable. Le micro-ordinateur doit donc fonctionner avec une tension d'alimentation aussi basse que 3,5 volts puisqu'il y a une perte de 1 volt environ entre l'entrée de la batterie, la régulation et le microprocesseur. Contrairement à la plupart des micro-ordinateurs existants en 5 volts, la fourchette de tension d'alimentation serait de 3,50 à 5,25 volts.

L'alimentation est du type zener car le courant normal du micro-ordinateur avoisine 55 mA et la perte de puissance de 0,25 watt seulement. La faible dissipation de chaleur prolonge la durée de vie, en particulier quand le micro-ordinateur fonctionne à une température élevée, juste avant que la ventilation du moteur ait pu équilibrer la température.

Si l'alimentation du micro-ordinateur + V_2 est reliée à celle de l'interface périphérique + V_1, en **figure 3,** on obtient deux niveaux de régulation. Les niveaux sont en série, ce qui est une protection supplémentaire contre les variations de tension. Cette disposition permet aussi au chip interface périphérique de détecter une chute de tension en C_1 avant que la tension en C_2 ne baisse notablement. Ainsi, un signal de contrôle peut être créé et envoyé au micro-processeur de façon à ce qu'il sauvegarde ses informations.

Le logiciel

Le programme d'instructions du microprocesseur est contenu dans une mémoire effaçable électriquement. En fonctionnement normal, ce programme d'instructions est un ensemble fixe de mots binaires. Lors de la fabrication du matériel et l'assemblage du module de contrôle, il serait souhaitable de placer des tests spécialisés dans la mémoire. Une impulsion de haute tension (d'environ + 25 volts) est appliquée à une broche de « programmation ». Grâce à deux signaux de contrôle on peut charger les codes séries dans la mémoire en 500 ms et tous les mots peuvent être testés pour vérification en 25 ms environ. Il est donc extrêmement important pour la réussite de cette application que les mémoires électriquement effaçables soient développées avec succès.

La chaîne de montage finale, en conséquence, peut stocker des programmes de contrôle moteur suivant un schéma de test standard jusqu'à ce que soit établie l'identité de la voiture. A ce moment, une sonde peut introduire les mots du code instruction.

La production des micro-ordinateurs

Le microprocesseur de contrôle moteur proposé demande un total de 12 broches : 6 pour l'alimentation, l'oscillateur local, la programmation et 6 supplémentaires pour l'échange de données.

Un tel circuit serait composé d'environ 20 000 transistors actifs sur un chip de 5 mm de côté. Un modèle de 14 broches laisserait 2 broches libres pour les contingences futures.

La personnalisation du circuit réduirait le nombre des interconnexions à un strict minimum.

Durant les opérations d'ingéniérie, il est souvent difficile de trouver assez de temps pour transformer un test d'instructions en un schéma ingénieux de boucles et de ramifications qui entrent dans l'espace de mémoire disponible.

Pour cette raison, la version prototype du chip micro-ordinateur devra probablement être assemblée dans un module à 40 broches de telle façon que l'adresse des données et les systèmes de contrôle puissent être amenés à travers environ 24 liaisons supplémentaires. Dans cette configuration, l'espace de mémoire disponible est doublé (2 048 à 4 098 mots).

Les concepts présentés ici pour la prochaine génération de micro-ordinateurs contrôlant totalement les moteurs automobiles ne sont pas encore adaptés à la production de masse. Néanmoins, ils montrent que l'on peut réaliser des systèmes peu coûteux en utilisant la technologie des semi-conducteurs.

Le besoin urgent des capteurs

Le manque de capteurs suffisamment précis et bon marché pose un problème sérieux dans le développement de systèmes de contrôle et de gestion de l'énergie automobile. Ces capteurs sont nécessaires pour réaliser l'interface entre la logique du système de contrôle et les variables du monde réel (par exemple température et pression). Ils ont été produits jusqu'à présent pour des applications industrielles et aérospatiales à des coûts élevés. Le problème du coût des capteurs sera résolu grâce à une étroite collaboration entre les concepteurs automobiles et les fabricants de semi-conducteurs.

L'industrie des semi-conducteurs a traité de gros marchés d'O.E.M. (Original Equipment Manufacturer) depuis plusieurs années et possède une technologie de production de masse à bas prix très bien développée. Pour cette raison, le marché du capteur est un élément de croissance pour elle. Néanmoins, tous les types de senseurs n'ont pas effectivement ou potentiellement un prix bas. Il faut donc tenter d'identifier les types du senseur qui sont applicables à la technologie des semi-conducteurs !

Capteurs de température

Ces équipements représentent le plus grand marché des capteurs simples de tous types et sont dominés par les thermocouples, les thermistors et les détecteurs de température.

Capteurs de pression

A de très rares exceptions ces capteurs déduisent la pression du déplacement d'un élément mécanique grâce à une différence de pression à travers cet élément. L'élément mécanique le plus couramment utilisé est le tube à diaphragme (un tube fermé et enroulé assujetti à une pression interne). Le déplacement de l'élément mécanique est mesuré directement par des méthodes potentiométriques, de répulsion, capacitive ou, indirectement, par l'utilisation de jauges de contraintes.

Grâce à la nature électromécanique du capteur de pression, sa réponse mécanique influence la performance du transducteur autant que l'élément lui-même. La plupart des tentatives d'amélioration des performances ont été centrées sur les méthodes employées, pour mesurer la variation de l'élément mécanique.

Capteurs de position

Le capteur de position est un domaine hautement diversifié qui utilise des technologies diverses. Les capteurs de position produisent des signaux de sortie analogiques ou digitaux. Le premier fournit un signal continu proportionnel à la position ; le second produit un signal digital fonction de positions fixes et prédéterminées. La mesure de la position peut être effectuée avec ou sans contact avec l'objet dont la position est à déterminer.

Mesure du débit d'un liquide

Comme les capteurs de pression, les capteurs de débit sont affectés par le flux du liquide. Les capteurs de vitesse ou de débit sont identiques et tous deux doivent être calibrés pour un fluide spécifique.

De tous les transducteurs, ceux adaptés aux fluides semblent être les moins susceptibles d'être conçus à l'aide de la technologie des semiconducteurs. On connaît un seul capteur de fluide qui utilise la technologie des semi-conducteurs. Il fonctionne sur le principe de la mesure d'une température induite par le flux entre 2 transistors appariés à l'entrée d'un amplificateur différentiel. Ce type de capteurs est exceptionnel, de plus, il est extrêmement sensible au fluide et à la température ambiante.

Capteurs d'environnement

Les senseurs d'environnement à l'état solide pour l'humidité et la composition des gaz sont parmi les plus récents et les plus mal compris. Beaucoup de ces senseurs sont de mauvaise qualité, ne sont pas fiables et sont utilisés la plupart du temps dans des buts qualitatifs. Malgré leur désavantages, ils sont bien adaptés à la technologie des semi-conducteurs. Par exemple les senseurs de gaz MOSFET utilisent la sensibilité bien connue des semi-conducteurs MOSFET aux contaminants comme principe de fonctionnement. La difficulté majeure avec ces senseurs est leur manque de spécificité car beaucoup de paramètres dûs à l'environnement peuvent les affecter d'une manière similaire. ■

Gene PUCKETT
est Directeur de stratégie commerciale dans le groupe semi-conducteur de MOTOROLA où il est responsable du planning stratégique et de la direction du programme des systèmes électroniques automobiles nouveaux et à venir.

John MARLEY
est membre de l'équipe technique du département CI de MOTOROLA où il s'occupe de systèmes destinés à l'automobile. Ces six dernières années, il a travaillé sur des questions de circuits intégrés à la demande pour l'automobile ainsi que de chips à processeurs centraux spécialisés.

John GRAGG
est directeur de projet pour le développement de la production de capteur à l'état solide au laboratoire de développement et de recherche des semi-conducteurs du groupe semi-conducteur de MOTOROLA. Il est chargé de constituer une équipe de projet qui entreprendra le développement de ses capteurs pour l'utilisation avec des systèmes de contrôle microprocesseurs.

NASCOM 1

ou le portrait d'un sujet britannique au-dessus de tout soupçon

COMME tout produit britannique le NASCOM 1, ordinateur personnel, possède des qualités typiquement anglaises. Il est racé, puissant, sophistiqué, étonnamment compétitif et raffiné.
Racé, parce qu'il descend directement d'une technologie éprouvée.
Puissant, parce qu'équipé du fameux micro-processeur Z 80.
Sophistiqué parce que ses auteurs l'ont conçu pour accepter, dès l'origine, toutes les extensions dont vous voudrez le doter ultérieurement : sortie vidéo, télétype, basic version mini ou étendue, RAM jusqu'à 64 K, floppy-disque, etc...
Compétitif, parce que comparé aux autres systèmes de sa classe, il reste très bon marché par les performances qu'il présente.
Raffiné, car comme tout bon anglais, il fait partie d'un club privé, dont, il vous ouvre toutes grandes les portes; vous y trouverez une bibliothèque de programmes déjà réalisés, mais rien ne vous empêchera d'y faire figurer les vôtres.
Ajoutez l'élégance, car vous pouvez l'habiller dans un rack d'aspect très professionnel, et, vous aurez mis la touche finale à ce portrait d'un sujet britannique au-dessus de tout soupçon.



Le système de base, se raccordant directement à un téléviseur standard et à un magnéto cassette, comprend le clavier alphanumérique, et, la carte de base en kit, avec :

- Micro-processeur Z 80
- Interface vidéo et TV
- Interface magnéto-cassette
- Inter face E/S série, boucle 20 MA ou RS 232
- Inter face parallèle 16 E/S
- RAM 1 K x 8 disponible utilisateur
- Moniteur NASBUG
- Connecteur d'extension

Livré avec une documentation détaillée dont 2 manuels de montage et de logiciel en français .... 2490 F/TTC

Options disponibles en kit

- Carte mémoire 8 K x 8
- Carte mémoire 16 K x 8
- Carte mémoire 32 K x 8
- Tiny Basic en EPROM
- Assembleur-éditeur
- Alimentation 2,2 A
- Carte Buffer-bus

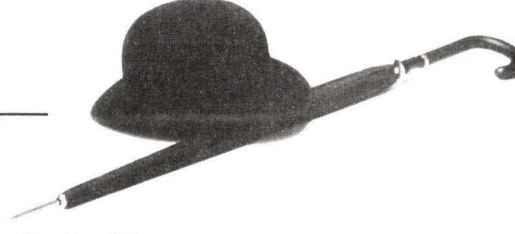
Options bientôt disponibles

- Rack 19"
- Carte E/S supplémentaire
- Carte relais
- Alimentation 8 A



Options en préparation

- Basic étendu
- Floppy-disque



carte mémoire 32 K x 8 **2980 F/TTC**

Liste des dépositaires

CHARLET 1, Cours du 14 Juillet - 47000 AGEN
CSE 15, rue Clovis - 57000 METZ
DELOCK 4, rue Colbert - 59000 LILLE
ELECTROME 17, rue Fondaudège
33000 BORDEAUX
EQUIP. ELEC. EST 7, rue de la Loi
68100 MULHOUSE
FANATRONIC 35, rue de la Croix-Nivert -
75015 PARIS
FANATRONIC 2, bd du Sud-Est - 92000 NANTERRE
LISCO 43, Grand-Place - 38000 GRENOBLE
REBOUL 34, rue d'Arènes - 25000 BESANÇON
SELECTRONIC 14, Bd Carnot - 59000 LILLE
SELFCO 31, rue du Fossé des Treize -
67000 STRASBOURG
SOS T.V. AYZE - 74130 BONNEVILLE
VALUNIT 63, rue Diderot - 42300 ROANNE



Nascom Microcomputers

IMPORTATEUR
JCS COMPOSANTS
35, rue de la Croix Nivert 75015 PARIS - Tél. 306.93.69

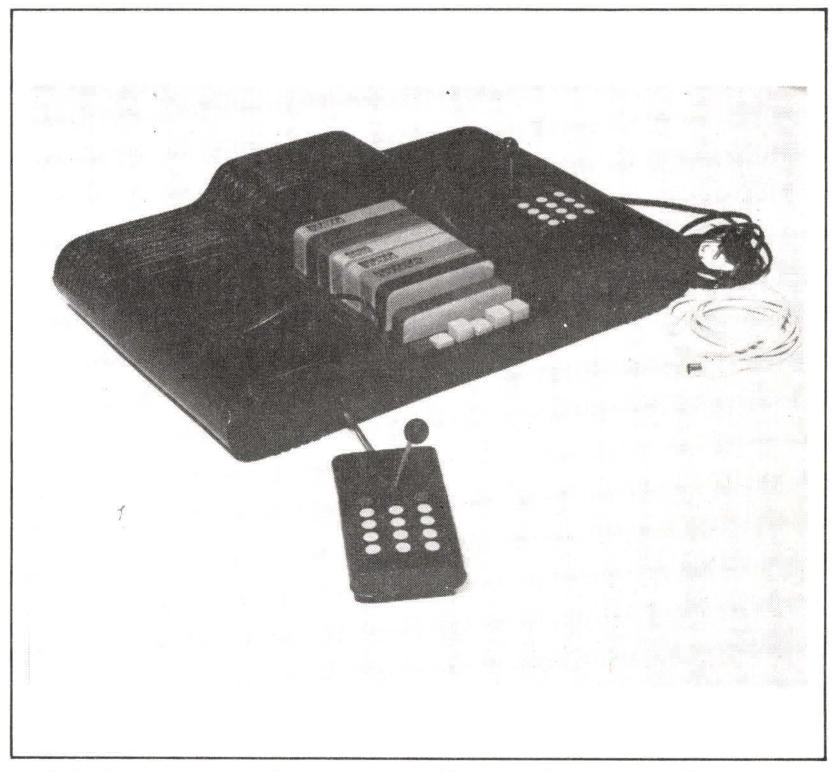
Circuits LSI pour jeux vidéo à microprocesseur

Les premiers jeux vidéo ont été commercialisés en 1972 et étaient réalisés soit à base de circuits analogiques tels que générateurs de rampes, différenciateurs, etc., soit de circuits logiques à niveau d'intégration faible ou moyen (SSI *, MSI *).

Les premiers circuits à haut niveau d'intégration (LSI *) étaient à leur tour introduits avec succès début 1976, avec un circuit spécialisé pour 4 jeux ; la diversité des jeux proposés restait cependant limitée.

L'avènement des microprocesseurs et leur introduction en tant qu'élément de base a provoqué un changement radical tant au niveau du matériel que des performances.

Non seulement le concept microprocesseur permet de s'affranchir de l'importante charge en temps et coût de développement nécessaire à la réalisation de circuits complexes spécialisés pour de nouveaux jeux, mais de plus il permet de proposer de véritables « jeux programmables » pratiquement sans limite, si ce n'est celle de l'imagination, en changeant uniquement le programme à exécuter par le microprocesseur.



« Vidéo Système » OC 2000 de la Société Occitane d'Electronique. Ce système est conçu autour des circuits 2650, PVI, USG.

Dans les jeux vidéo à base de microprocesseur, l'écran de télévision est utilisé comme une matrice de points. Deux méthodes de générateur d'images peuvent être distinguées :

- Le concept « RAM » dans lequel une mémoire RAM tampon contient les données nécessaires à la visualisation d'une image complète.
- Le concept « OBJET » dans lequel plusieurs petites mémoires tampon contiennent uniquement les données relatives aux objets à visualiser. Une circuiterie de contrôle programmable prend en charge le positionnement des objets sur la matrice-écran.

Dans les deux cas, la sortie vidéo peut gérer directement le faisceau d'électrons du tube d'image (« jeux résidents ») ou le module jeu peut envoyer les données à travers un modulateur sur le câble d'antenne du récepteur.

** SSI : Small Scale Integration, intégration à faible échelle.*

** MSI : Medium Scale Integration, intégration à moyenne échelle.*

** LSI : Large Scale Integration, intégration à grande échelle.*

Le concept « RAM »

Une mémoire vive (RAM) joue le rôle de « tampon d'écran » et contient toutes les informations nécessaires à une image complète, c'est-à-dire que la définition recherchée ainsi que le nombre de couleurs déterminent la capacité de mémoire nécessaire. Le schéma de principe d'un système de jeu utilisant cette technique est donné **figure 1.**

Par exemple si l'écran est considéré comme une matrice de 128 x 128 points et que l'on désire 4 couleurs (codées sur 2 bits) la capacité mémoire nécessaire est de $2^7 \times 2^7 \times 2^1 = 32\ 768$ bits (32 K bits). Cette mémoire est accessible par le microprocesseur pour composer l'image et par le canal d'accès direct mémoire (DMA) pour extraire les données à afficher à un moment donné sur l'écran.

Les composants mémoires les mieux adaptés (les moins chers) sont les mémoires RAM dynamiques de grande capacité. On peut les utiliser ici de façon astucieuse puisque le rafraîchissement est réalisé automatiquement par lecture permanente des données par le canal DMA.

Séquencement et contrôle

Le « système de séquencement » assure le lien entre un emplacement dans la mémoire tampon et la position correspondante sur l'écran T.V en fournissant les adresses mémoire et les signaux de synchronisation adéquats.

De cette façon, le canal DMA peut utiliser ces adresses pour lire les données dans la RAM en synchronisme avec l'exploration du tube T.V. Ces données, en combinaison avec les signaux de syn-

Le circuit d'interface vidéo programmable ou P.V.I. se comporte vis-à-vis du microprocesseur comme une zone mémoire.

chronisation sont converties, en un signal vidéo, ou sont utilisées pour la commande directe du tube image.

De plus, tous les signaux de séquencement pour le canal DMA et le micro-ordinateur sont fournis par ce « système de séquencement ».

Le micro-ordinateur

Le jeu à exécuter est complètement indépendant de la position des objets dans la mémoire tampon. En fait, le microprocesseur ayant accès à la mémoire c'est le programme qu'il exécute qui déterminera le jeu lui-même. Apparitions, mouvements, rotations, etc., des objets sont réalisés en stockant des ensembles appropriés de bit dans la mémoire RAM tampon.

Ces fonctions peuvent également être affectées par des commandes externes telles que potentiomètres, commutateurs ou manipulateurs actionnés par les joueurs.

Conclusion concernant le concept RAM

Le concept RAM permet une grande souplesse dans les jeux à exécuter puisque tous les points de l'écran peuvent être contrôlés individuellement. Toutes formes d'objet et composition de couleurs sont théoriquement possibles.

En contrepartie cela impose d'agir sur les objets au niveau des bits (ou de groupes de bits) ce qui se traduit rapidement par des programmes de gestion relativement complexes.

Par ailleurs, le microprocesseur est très chargé puisqu'il a à mettre à jour en permanence tout le contenu de la RAM tampon. En fait, les possibilités sont limitées par le nombre d'objets en mouvement, leur vitesse et les chances de collision qui peuvent être supervisées par le microprocesseur.

Le concept RAM impose également un nombre relativement important de boîtiers et le coût est donc assez élevé.

Ce coût n'est acceptable que dans la mesure où son intérêt réside surtout dans ses possibilités d'extension à d'autres applications que les jeux tels que terminaux de visualisation, système d'évaluation.

Fig. 1. – Jeu vidéo à microprocesseur ; le concept « RAM ». La mémoire vive joue le rôle de « tampon d'écran » et contient les informations nécessaires à une image complète.

Le concept « Objets »

Le principe de fonctionnement du système basé sur le concept « Objet » est donné **figure 2.**

Dans l'approche « Objets » proposée ici, le microprocesseur Signetics 2650 lit le programme-jeu stocké en mémoire ROM, et contrôle la vidéo présentée au récepteur T.V. Un circuit d'interface spécial P.V.I. (Programmable Video Interface) programmable par l'unité centrale interprète les commandes du microprocesseur et présente les signaux vidéo au circuit D.V.S. (Digital Video Summer) qui est un codeur fournissant le signal vidéo et des signaux de séquencement en provenance d'un générateur de synchronisation (USG Universal Sync. Generation).

L'unité centrale communique avec le P.V.I. par l'intermédiaire des bus d'adresses et de données. Le P.V.I. se comporte vis-à-vis du

microprocesseur comme une zone mémoire dans laquelle le microprocesseur peut écrire des données, le P.V.I. fournit ensuite les signaux vidéo en fonction des informations stockées dans sa zone RAM.

Par ailleurs, le P.V.I. présente au microprocesseur des informations concernant son état propre et l'état des dispositifs d'entrée/sortie (ex. : Collision entre objets, etc.). Pour ce faire, le P.V.I. écrit des informations dans certains emplacements mémoire, emplacements qui peuvent être lus par le microprocesseur afin de prendre des décisions.

Description du P.V.I.

La **figure 3** représente l'architecture d'un P.V.I.

Le P.V.I. est un circuit à haut niveau d'intégration réalisé en MOS canal N. Les communications entre le microprocesseur 2650 et le P.V.I. sont assurées par les bus du système (voir encadré).

L'interface vidéo programmable contient les éléments nécessaires pour :

- décrire la forme, la couleur, la position et les dimensions de quatre objets ;
- décrire la forme et la couleur du fond ;
- engendrer le son et le score ;
- décrire les collisions entre objets ou entre objets et le champ du jeu, la fin d'affichage d'un objet et la fin d'une image ;
- mémoriser la dernière valeur de conversion A/D des entrées (potentiomètres) ;
- effectuer des calculs grâce à une zone mémoire « bloc notes ».

Le mécanisme de contrôle des objets est le suivant : on associe à chaque objet un petit bloc-mémoire RAM et une paire de registres de coordonnées qui indiquent l'emplacement de l'écran où l'objet doit être visualisé. Un ensemble de compteur (en vertical et en horizontal) indiquent en permanence où se trouve le point d'impact du faisceau d'électrons. Lorsqu'il y a coïncidence entre la position du faisceau et les registres de coordonnées, l'objet en question est affiché.

Les mouvements des objets sont obtenus en programmant le microprocesseur de façon adéquate pour modifier les coordonnées des objets. Un déplacement vertical est obtenu simplement en incrémentant ou décrémentant le registre de coordonnée verticale. Un déplacement horizontal en

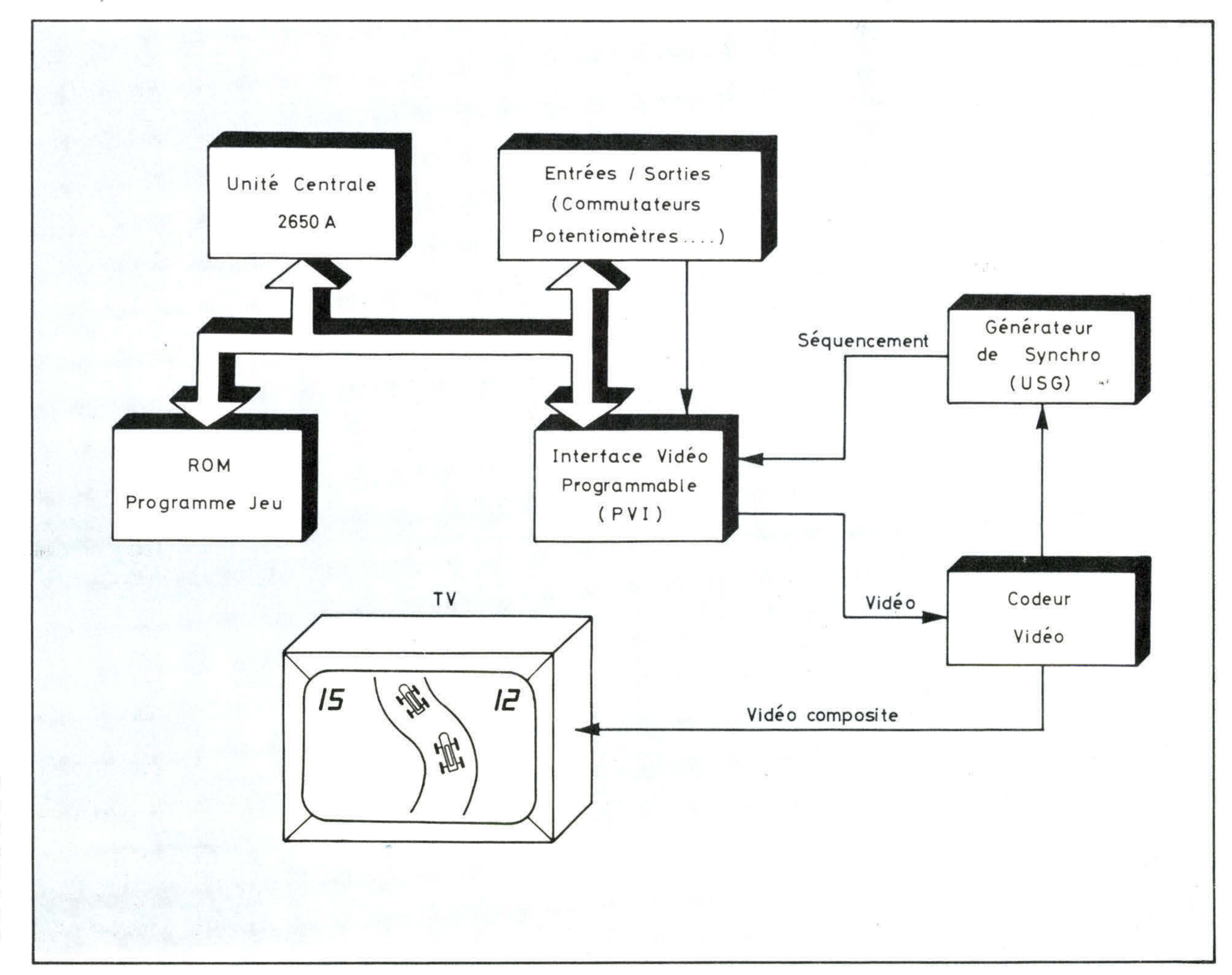


Fig. 2. – Jeu vidéo à microprocesseur : le concept « objet ». Ici, le microprocesseur lit le programme de jeu stocké en mémoire ROM et contrôle la vidéo présentée au récepteur T.V.

Le P.V.I. contient une mémoire RAM de 256 octets qui stocke les informations concernant la définition des objets et du champ de jeu.

incrémentant ou décrémentant le registre de coordonnées horizontales.

De cette manière chaque objet possède deux degrés de liberté.

Fonction du générateur de synchronisation USG

Ce boîtier fournit la fréquence d'horloge de base ainsi que les signaux de remise à zéro en horizontal et vertical du P.V.I. Les compteurs, horizontal et vertical, de l'USG fournissent au P.V.I. une représentation en coordonnées cartésiennes de l'écran de T.V.

Conclusion sur le concept « Objets »

Dans ce cas la composition d'un objet est réalisée de façon similaire à l'approche « RAM » en stockant la configuration de l'objet en mémoire RAM du P.V.I. Par contre, le déplacement d'un objet, la modification de sa grandeur ou la détection des collisions se font beaucoup plus simplement en transférant les nouvelles coordonnées et les mots de contrôle adéquats entre le microprocesseur et le P.V.I.

Ceci permet d'une part de décharger considérablement l'unité centrale qui n'a pas à remettre à jour l'ensemble de l'écran et peut donc se consacrer à plus d'objets et minimiser les tailles des programmes.

De plus, le concept « Objets » réduit de façon considérable le nombre de composants à mettre en œuvre, donc le coût global du système, tout en permettant une grande variété de jeux en changeant uniquement la ROM contenant le programme spécifique à chaque jeu. ■

H. MAGRO *

** H. MAGRO est responsable des produits « Microprocesseurs » à la R.T.C.*

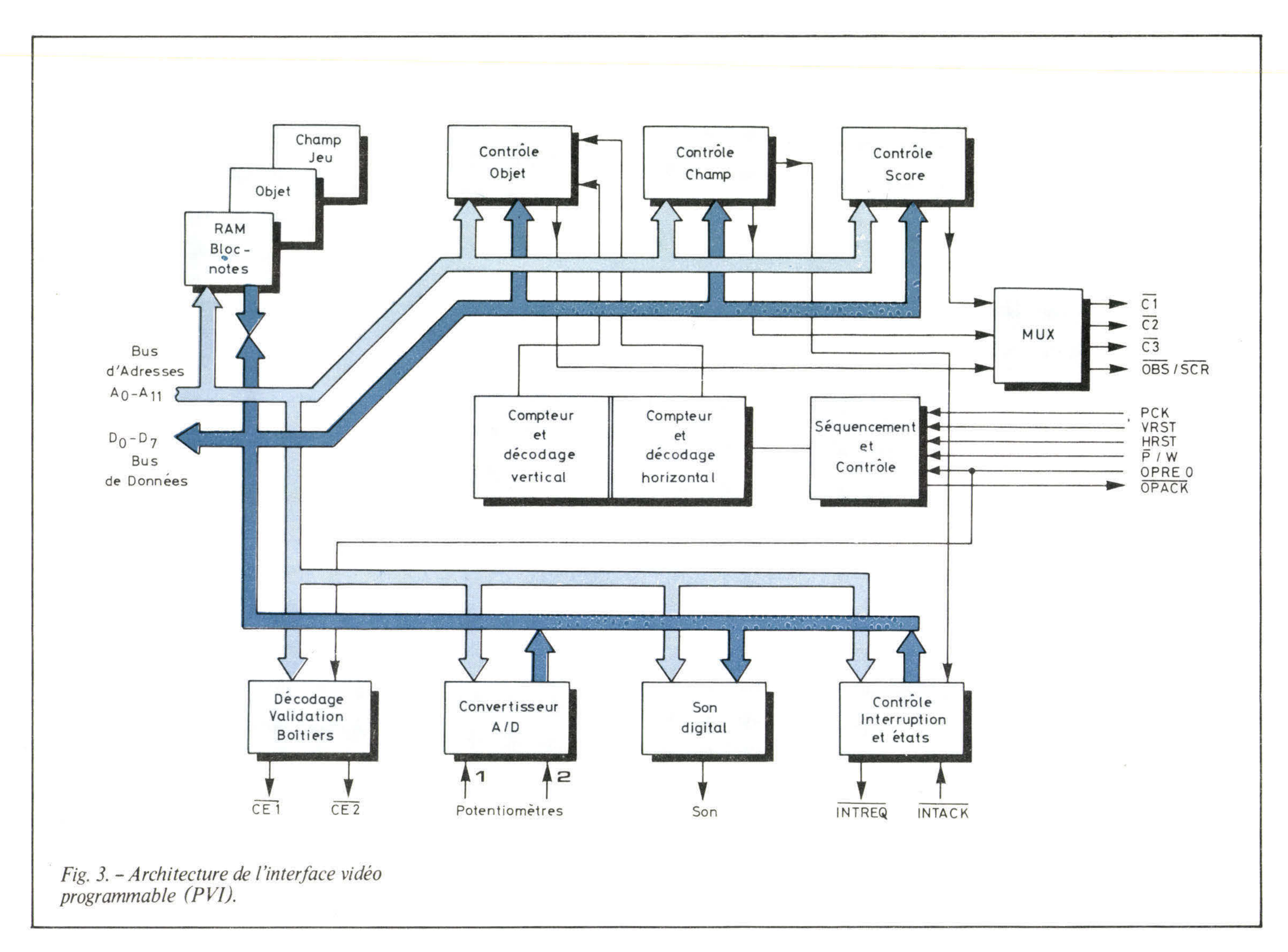


Fig. 3. – Architecture de l'interface vidéo programmable (PVI).

Le P.V.I., circuit d'interface vidéo-programmable

Le P.V.I. (Programmable Video Interface) fait partie de la catégorie des circuits d'interfaces programmables dont l'apparition sur le marché est le résultat de deux éléments essentiels :

- Les progrès technologiques extraordinaires réalisés ces dernières années dans le domaine de l'intégration à haut niveau, permettent de réaliser des circuits intégrés de très grande complexité.
- L'introduction du microprocesseur (en fait une conséquence du phénomène précédent) qui a permis de rendre accessible à tous les domaines d'application de l'électronique des techniques propres à l'informatique et en particulier l'organisation d'un système autour d'une Unité centrale (le microprocesseur), dialoguant avec l'environnement par l'intermédiaire de « BUS » (d'adresse, de donnée et de contrôle).

Avec le microprocesseur apparaissait un concept nouveau pour l'électronicien : le « logiciel » (c'est-à-dire la suite des instructions à faire exécuter par l'unité centrale et contenue dans la mémoire de programme). Est également apparue une autre préoccupation, celle de la répartition des tâches entre « matériel » et « logiciel », soit pour exécuter des opérations à des vitesses incompatibles avec les possibilités du microprocesseur (association d'opérateurs en logique câblée) soit pour décharger l'unité centrale de tâches répétitives et bien connues pour un domaine d'application donné.

C'est ce second aspect qui a donné en particulier naissance à des circuits d'environnement microprocesseur tels que le 2651 PCI et 2652 MPCC particulièrement adaptés au domaine des communications de données.

De même que ces deux composants, le P.V.I. est un dispositif orienté « BUS » et dont le mode de fonctionnement est programmé par l'unité centrale. Le microprocesseur 2650 a accès aux principaux éléments fonctionnels du P.V.I. par l'intermédiaire des bus d'adresse et de donnée : il sélectionne l'élément avec lequel on peut communiquer grâce au bus d'adresse et l'information est véhiculée dans un sens ou dans l'autre par le bus de donnée.

Le P.V.I. contient une mémoire RAM de 256 octets qui stocke les informations concernant la définition des objets et du champ de jeu. Le 2650 et le P.V.I. ont tous deux accès à cette information. La logique de contrôle du P.V.I. examine l'information stockée dans la mémoire, par le microprocesseur (ex. : placer l'objet nº 1 à l'emplacement H.120, V.37), et fournit les signaux vidéo adéquats à l'emplacement prévu. Les signaux de sortie comprennent les couleurs des objets, du fond et du score et une fréquence audible.

Le bloc de conversion A/D transforme les informations analogiques de position du potentiomètre en données binaires acceptables par le microprocesseur.

Le bloc « son digital » est constitué d'un générateur d'onde carrée dont la fréquence de sortie est contrôlée par le microprocesseur.

Le bloc « Etats et contrôle d'interruption » permet au P.V.I. d'une part d'interrompre le 2650, d'autre part de lui fournir des informations concernant le déroulement du jeu (ex. : collisions).

La mémoire interne du P.V.I. (256 octets) contient la description de 4 objets et de leurs éventuelles reproductions, la description du « champ de jeu », des informations de contrôles et d'entrées/sorties ainsi que des zones « bloc-notes » pour des calculs.

Génération de la vidéo objet

— Chaque objet est défini par 5 données :

- Forme de l'objet : sur 80 bits.
- Position de l'objet : sur 16 bits.
- Position d'un double de l'objet : sur 16 bits.

(Ces trois éléments constituent la description de l'objet).

- Dimensions de l'objet : sur 2 bits.
- Couleur de l'objet : sur 3 bits.

Contrôles et entrées/sorties

Parmi ces emplacements mémoires certains sont écrits par l'unité centrale de façon à programmer le P.V.I. :

- Dimensions des quatre objets : 4 x 2 bits.
- Couleurs des quatre objets : 4 x 3 bits.
- Format et position du score : 2 bits.
- Couleurs du fond du jeu (écran), du champ de jeu et validation : 2 x 3 bits + 1 bit.
- Son : 8 bits.
- Valeur des 4 digits de score : 4 x 4 bits.

D'autres emplacements mémoires concernent des informations d'états lues par l'unité centrale pour prendre des décisions ou effectuer des calculs.

- Collision entre objets et champ de jeu : 4 bits.
- Collisions entre objets : 6 bits.
- Fin d'affichage des objets : 4 x 1 bit.
- Résultats conversion A/D des deux potentiomètres : 2 x 8 bits.



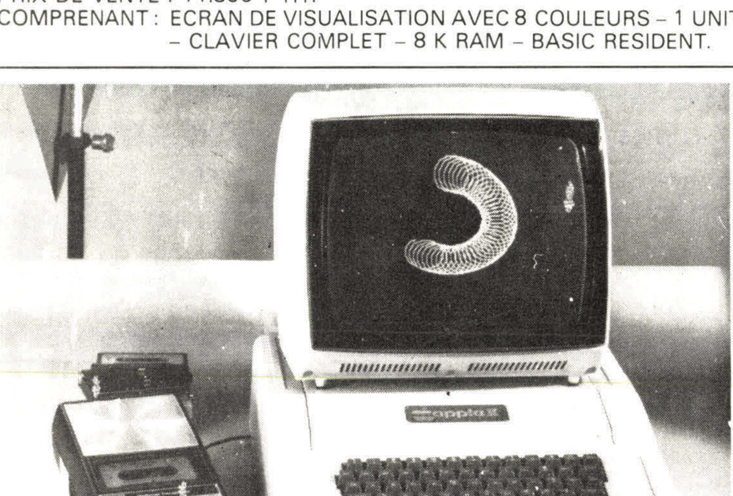
Informatic Systèmes TéléCom

DISTRIBUTEURS RECHERCHÉS DANS TOUTE LA FRANCE

MICRO-ORDINATEUR COMPUCOLOR

- MICROPROCESSEUR 8080 A.
- MEMOIRE RAM DE 8 A 32 Ko.
- BASIC 16 K RESIDENT
- INTERFACE RS 232 C.
- ECRAN DE VISUALISATION (64 c x 16/32).
- 8 COULEURS.
- TRACE GRAPHIQUE.
- 1 UNITE DE MINI-DISQUETTE.
- 64 CARACTERES SPECIAUX.
- OPTION IMPRIMANTE.

PRIX DE VENTE : 11.800 F HT.
COMPRENANT : ECRAN DE VISUALISATION AVEC 8 COULEURS – 1 UNITE DE MINI-DISQUETTE INTEGREE – CLAVIER COMPLET – 8 K RAM – BASIC RESIDENT.



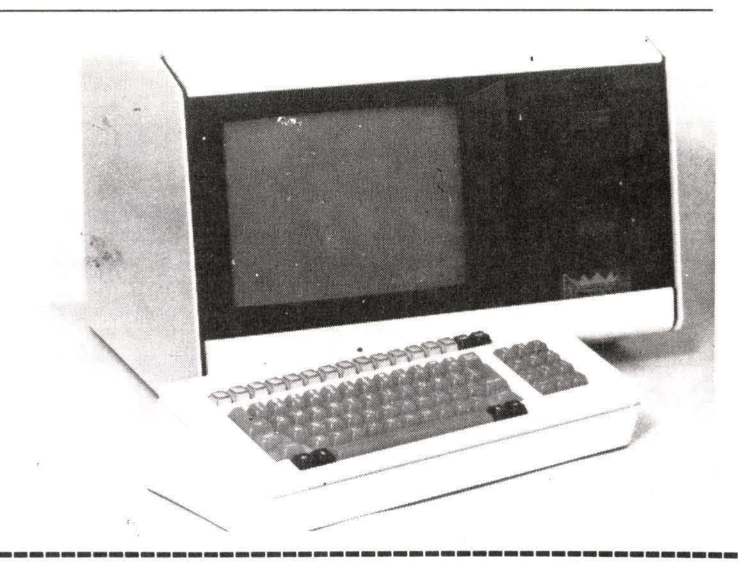
MICRO-ORDINATEUR APPLE-II

- MICROPROCESSEUR ROCKWELL 6502 - RAM EXTENSIBLE DE 4 K A 48 K.
- BASIC - MONITEUR - ASSEMBLEUR - DESASSEMBLEUR (ROM).
- SORTIE VIDEO 24 LIGNES/40 COLONNES.
- GRAPHIQUES FINS EN COULEURS SUR T.V. (RVB-SECAM).
- INTERFACES MAGNETOPHONE ET ENTREES ANALOGIQUES - HAUT-PARLEUR INCORPORE.
- 8 PERIPHERIQUES CONNECTABLES DONT :
  - IMPRIMANTE, MODEM, CARTE DE COMMUNICATION RS 232
  - CARTE DE RECONNAISSANCE VOCALE (32 MOTS QUELCONQUES)
  - FLOPPY DISQUES (1 A 14 FOIS 116 Ko).

* DOS : FICHIERS DE DONNÉES EN ACCÈS SÉQUENTIEL INDEXÉ PROGRAMMATHIQUE/CHAINAGE DES PROGRAMMES/PROTECTIONS D'ÉCRITURE.

MICRO-ORDINATEUR I.S.T.C. 5000

- MICROPROCESSEUR Z80 - RAM de 32 K à 64 K.
- SORTIE VIDEO 24 LIGNES/80 COLONNES.
- GENERATEUR DE CARACTERES PROGRAMMABLE.
- 1 OU 2 MINI-FLOPPY DISQUES (DOUBLE FACE) INTEGRES.
- DOS-EDITEUR DE TEXTE.
- MACRO ASSEMBLEUR.
- BASIC ETENDU (IF THEN ELSE, WHILE, PRINTUSING).
- FORTRAN IV ANSI
- EDITEUR DE LIENS POUR MODULES FORTRAN.
- 2 A 5 CONNECTEURS BUS S-100.
- INTERRUPTIONS CHAINEES AVEC PRIORITES (8 NIVEAUX).
- CARTE DE COMMUNICATION (SYNCHRONE/ASYNCHRONE).
- IMPRIMANTE AVEC INTERFACE.



Micro-Systèmes

Bon réponse à retourner à : I.S.T.C., 7 à 11, rue Paul-Barruel, 75015 Paris. Tél. : 306.46.06.

Raison Sociale ....

Activité ....

Nom et Fonction ....

....

Intéressé par : COMPUCOLOR ☐
APPLE II ☐ I.S.T.C. 5000 ☐

Adresse ....

....

Tél ....

☐ UNE DEMONSTRATION ☐ LA VISITE D'UN COMMERCIAL

Le BASIC

Avant d'aborder l'étude particulière, que nous vous proposerons dans le prochain numéro de « Micro-Systèmes », des listes, fichiers et tableaux nécessaires à la gestion de stocks, à la mise à jour d'un fichier d'adresses et en général aux différentes manières de procéder aux tris, insertions ou radiations indispensables à l'optimisation d'un programme de traitement (maximum de résultats pour un espace mémoire occupé et un temps d'exécution minimum), nous pensons qu'il peut être intéressant pour vous d'approfondir et de reprendre plus en détails les instructions que nous avons décrites jusqu'à présent.

Ainsi, nous analyserons les instructions PRINT ; PRINT ON, OFF ; PRINT USING ; LET ; READ et DATA ; IF-GO TO ; IF-THEN ; FOR et NEXT pour parvenir aux notions de boucles, de boucles imbriquées et de sauts.

Vous trouverez, en encadré, une série de questions relatives à la programmation en langage BASIC auxquelles nous vous invitons à répondre. La solution détaillée de chacune de ces questions figure en conclusion de cet article.

« La Vierge et l'Enfant »
Célèbre illustration ; réalisée par la revue américaine « Creative Computing », et programmable en Basic sur micro-ordinateur.

PRINT

Cette instruction produit l'impression d'un texte. Plusieurs syntaxes peuvent être utilisées sous les formes suivantes :

```
100 PRINT (expression ou varia-
          ble 1), (expression
          ou variable 2) ;
110 PRINT
120 PRINT (expression ou varia-
          ble 1), (expression
          ou variable 2) ;
130 PRINT « TEXTE », A
```

Sur les machines disposant de plus d'un canal d'entrée/sortie, on rencontre également des directives

En BASIC, toutes les variables sont automatiquement initialisées à zéro.

de sélection des périphériques tel que :

```
140 PRINT ON (ou OFF)
```

100 On obtient l'impression de la valeur calculée de l'expression 1, suivie de celle de l'expression 2, sous forme purement numérique. Le ; en fin de ligne signifie que la prochaine impression se fera à la suite de la première, sur la même ligne.

110 L'instruction vide sert à créer un simple passage à la ligne suivante.

120 Un ordre d'impression non-muni de « ; » (ou de « , » sur certaines machines) à la fin de l'instruction, produira le passage automatique à la ligne suivante.

130 Chaque fois que l'on rencontre un texte entre guillemets, il est imprimé dans son intégralité, suivi de la valeur de l'expression ou de la variable. Si l'on est en présence d'une suite de textes, expressions ou variables à calculer de la forme :

```
150 PRINT « TEXTE 1 », A,
    « MESSAGE à paraître tel
    quel », B, « TEXTE 3 »
```

Cet entrelacement de texte et de valeurs à calculer facilite la lecture de résultats.

Exemple

```
10 F = 10
20 PRINT « LE SOLDE DU
   COMPTE DE M. DURAND
   EST DE », F, « FRANCS »
30 END
```

fait apparaître le message suivant :

```
LE SOLDE DU COMPTE DE
M. DURAND EST DE
10 FRANCS
```

PRINT USING

Cette instruction d'impression utilise une autre instruction qui spécifie le format.

Exemple :

```
100 PRINT USING 200 « Nom du Fournisseur », N, I, J
200 : « - - - - - - - - - - - - - - - - - - - - » # # # . # #   # #   # #
```

La ligne 200 dans cet exemple est une ligne-image donnant le format de l'impression, tant en positionnement du texte qu'en taille et forme de nombres. Précisément, après le texte, le nombre N sera affiché à l'endroit des # en trois chiffres suivis de deux décimales (même s'il est égal à 3,141 il apparaîtra sous la forme bb3.14). A deux blancs de distance, suivra la valeur de I, à deux chiffres et pareillement pour J.

Rappelons que dans d'autres langages l'impression est distincte du calcul ; ne peuvent être imprimées que des variables dont la valeur est connue avant l'ordre PRINT.

LET

10 LET variable = expression ou variable ou valeur numérique.

Instruction d'affectation. Elle n'indique pas une égalité algébrique et ordonne à la machine d'effectuer les opérations demandées à droite du signe égal et d'assigner le résultat à la variable se trouvant à gauche du signe égal. « LET » signifie « SOIT » en français. Très souvent cette instruction est implicite et les affectations peuvent prendre les formes équivalentes suivantes :

```
50 LET X = A + B + C
```

ou simplement :

```
50 X = A + B + C
60 LET Y = 12
```

ou

```
60 Y = 12
```

En aucun cas l'opération d'affectation n'inclut un calcul implicite et la relation :

```
20 LET Y = 2 * Y ↑ 2 + 5
```

ne donne pas la solution de l'équation de second degré :

$$2\,y^2 - y + 5 = 0$$

Elle ne fait qu'affecter à la variable Y la valeur que prendrait l'expression $2\,y^2 + 5$, pour une ancienne valeur de y. Si, en rencontrant l'instruction 20, Y a la valeur 2, après l'exécution de l'instruction 20, Y aura la valeur

$$Y = 2 \,.\, 2^2 + 5 = 13.$$

READ et DATA

L'instruction READ sert à assigner aux variables qui sont indiquées, les valeurs numériques figurant dans une instruction DATA.

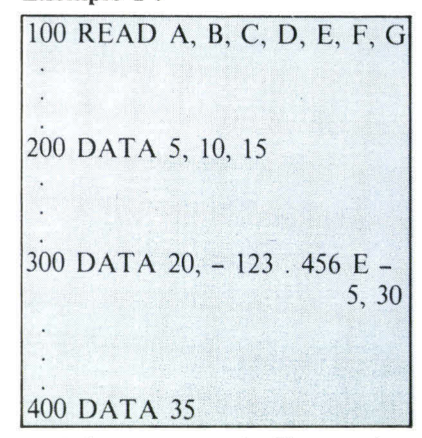
La première variable à lire (READ) sera la première valeur numérique de la liste de l'instruction DATA.

Les READ figurent généralement vers le début du programme, les DATA étant placés à une place arbitraire, habituellement avant l'instruction END.

Si l'on demande à lire un nombre de variables supérieures à celles qui figurent dans une instruction DATA, et s'il n'existe pas dans le programme une autre instruction DATA contenant le nombre manquant de données, un message d'erreur apparaîtra. Les instructions READ et DATA peuvent apparaître à deux ou trois reprises. Seul compte l'ordre d'apparition et le nombre total de variables.

Ces instructions permettent à un même programme d'être utilisé pour plusieurs applications si, seules, les variables d'entrées sont différentes ou lorsque les données sont inconnues lors de la conception du programme.

Exemple 1 :



```
100 READ A, B, C, D, E, F, G
 .
 .
 .
200 DATA 5, 10, 15
 .
 .
 .
300 DATA 20, - 123 . 456 E -
                           5, 30
 .
 .
 .
400 DATA 35
```

A la rencontre de l'instruction 100 on assignera à la variable A la valeur 5, à la variable B la valeur 10,... jusqu'à G qui prendra la valeur 35.

Voici une autre utilité du READ, dans un programme qui calcule la moyenne arithmétique de 5 nombres spécifiés en DATA, par l'intermédiaire d'une boucle :

Exemple 2 :

```
10 pour I = 1 à 5
20 READ A
30 S = S + A
40 boucle I
50 PRINT « LA MOYENNE ARITHMETIQUE EST », S/5
60 DATA 53, 71, 1E + 5, - 2,32 E + 4, 18, 21, 54, 32, etc.
70 END.
```

En BASIC toutes les variables sont automatiquement initialisés à zéro. Quand, à la première lecture on rencontre S, il prend la valeur 0.

Comme nous pouvons le constater, par cinq fois, grâce à la technique de la boucle, on additionnera des valeurs consécutives lues dans la liste de valeurs numériques de l'instruction DATA. Première remarque : il y a plus de valeurs dans les DATA que nécessaire. Ceci n'est pas une faute et ne génère pas de message d'erreur.

Les valeurs consécutives s'additionnent dans « S ». Un calcul simple dans le PRINT (S/5) donne alors la moyenne arithmétique.

La même instruction READ permet aussi la lecture des **tableaux.**

Exemple 3 :

```
 80 tableau T(5)
 90 pour k = 1 à 5
100 READ T(K)
110 boucle K
.
.
.
300 DATA 1, 10, 15, 2, 3, etc.
```

L'instruction DATA peut également contenir des **chaînes de caractères.**

10 DATA « COUCOU, Z », 1, 2 est permis.

READ A, B, C, D produira l'affectation des valeurs numériques correspondant aux codes ASCII de COUCOU et de Z aux variables A et B et des valeurs numériques 1 et 2 aux C et D.

GO TO

Cette instruction renvoie la machine à une ligne de programme donnée. Elle a plusieurs formes :

```
100 GO TO numéro de ligne
110 GO TO n1, n2, ... nm ON
                  expression.
```

100 — Cette forme est un simple branchement inconditionnel.

```
 50 . . . . . . . . . .
 .
 .
 .
100 GO TO 50
```

Après la ligne 100, le programme commencera à exécuter l'instruction contenue à la ligne 50.

L'instruction 100 GO TO 100 représente une boucle sans fin.

110 — Sous cette forme le GO TO ressemble à un aiguillage. Il y a saut à l'adresse n_i, si la partie entière de l'expression calculée est égale à n_i.

Exemple :

```
10 . . . . .
20 . . . . .
30 . . . . .
40 . . . . .
50 GO TO 10, 20, 30, 40
        ON N = N + 10
```

Ce programme contient un saut aux lignes 10, 20, 30 et 40 en fonction des valeurs de N.

Au premier passage, si N = 0 en initialisation, il y a saut à l'instruction d'adresse 10. Au deuxième passage par la ligne 50, N vaudra 10 et le saut se fera à l'adresse 20, etc.

IF - GO TO

Saut conditionnel :

```
IF (expression arithmétique) (relation) (expression arithmétique)
GO TO (numéro de ligne)
```

Si la relation logique est vraie il y a saut au numéro de ligne spécifié par l'instruction GO TO. Sinon, on continue en séquence :

```
10 IF X<=Y + 1 GO TO 100
15 . . . . .
```

Si, à la ligne 10, X satisfait la relation, le programme continue à l'adresse 100. Sinon, l'instruction 15 est exécutée.

Une forme courte de cette instruction est la suivante :

```
IF(variable ou expression arithmétique) GO TO (numéro de ligne).
```

Exemples :

```
10 IF X GO TO 100
15 IF X + Y ↑ 2 GO TO 100
```

Sous cette forme on exécute la ligne 100 **si X est différent de zéro.** Cette forme courte est équivalente à

```
10 IF X < > 0 GO TO 100
15 IF X + Y ↑ 2 < > 0 GO
                 TO 100
```

IF - THEN

Exécution conditionnelle d'une instruction ou d'un saut. La forme la plus générale de cette instruction est :

```
IF (expression arithmétique)
   (relation) (expression arith-
   métique) THEN (numéro
   de ligne ou instruction)
```

Exemples :

```
15 IF ABS (X) < = N THEN 40
22 IF X THEN PRINT « VA-
   LEUR NON-NULLE », X
28 IF Y > = SIN (X) THEN GO
   TO 40.
```

L'adresse de saut peut ou non être accompagnée d'un « GO TO ».

Dans la première instruction, la valeur absolue de X est comparée à N et si elle est plus petite ou égale à celle-ci il y a saut à la ligne 40.

Dans la seconde instruction, une valeur non-nulle de X, valide la condition et produit l'impression

Lorsque la raison d'une progression à une valeur négative, il y a exécution de la boucle tant que la variable est supérieure ou égale à la valeur finale.

du texte accompagnée de la valeur de X.

La troisième instruction, équivalente à la première, est plus proche du langage parlé.

FOR et NEXT

Ces instructions permettent de réaliser aisément des boucles. Prenons l'exemple suivant.

Calcul de la somme d'une progression arithmétique :

Pour X variant de 1 à 10, effectuer

```
S = S + X.
```

On peut procéder par les moyens déjà connus :

```
10 X = 1
20 S = S + X
30 X = X + 1
40 IF X < 11 THEN 20
50 END.
```

L'instruction FOR permet d'éviter l'emploi de ce programme hybride utilisant l'incrémentation de X et le test de passage par sa valeur extrême et donne lieu à l'exécution répétitive du programme pour diverses valeurs de X. Cela devient ainsi :

```
10 FOR X = 1 TO 10
20 S = S + X
30 NEXT X
40 END
```

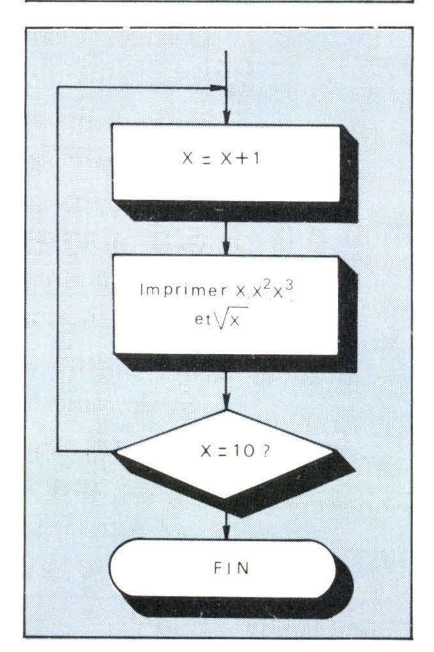


Fig. 1. – Organigramme correspondant à une boucle de programme.

La partie de programme à recommencer pour une autre valeur de X est délimitée par l'instruction NEXT.

Dans le cas général la valeur de X sera augmentée d'une quantité quelconque, la raison de la progression pouvant être positive ou négative et pas nécessairement égale à l'unité.

Le pas de cette progression (Step en anglo-saxon) est alors indiqué de la façon suivante :

```
10 FOR X = 1 TO 10 STEP 2
```

Ceci fera prendre à X les valeurs 1, 3, 5, 7, 9.

L'instruction :

```
10 FOR X = 10 TO 1 STEP-1
```

produira le déroulement des opérations dans l'ordre décroissant de X.

Si la valeur initiale est supérieure à la valeur finale, ou inférieure lorsque la raison de la progression est négative, il n'y aura pas d'exécution de la boucle et l'instruction suivant immédiatement le NEXT sera exécutée.

Les BOUCLES

Technique courante en programmation, une boucle s'utilise chaque fois qu'un programme comporte une ou plusieurs portions se répétant un certain nombre de fois avec ou sans changement de valeurs numériques.

Soit par exemple un programme qui se propose d'imprimer à la suite les 10 premiers nombres naturels, leur carré, leur cube et leur racine carrée. Si l'on n'utilise pas de boucle, le programme aura 11 lignes et sera conçu de la façon suivante :

```
10 PRINT 1, 1 * 1, 1 * 1 * 1, SQR (1)
20 PRINT 2, 2 *  2, 2 * 2 * 2, SQR (2)
30 PRINT 3, 3 * 3, 3 * 3 * 3, SQR (3)
.
.
.
100 PRINT 10, 10 * 10, 10 * 10 * 10, SQR (10)
110 END
```

En général, la forme de cette expression est la suivante :

```
FOR (variable) = (valeur initiale)
TO (valeur finale) STEP (raison)
```

Lorsque la raison de la progression a une valeur positive, il y a exécution de la boucle tant que la variable est inférieure ou égale à la valeur finale.

Lorsqu'elle est négative, il y a exécution de la boucle tant que la variable est supérieure ou égale à la valeur finale.

La forme générale d'une boucle FOR-NEXT peut contenir des expressions mettant en jeu des formules plus ou moins complexes pour la valeur initiale, la valeur finale et la raison de la progression :

```
FOR X = Y/Z * EXP (– W ↑
        1.234) TO (A – B)/3
        STEP (D – 5 * 6)/7
```

Une boucle, respectant **l'organigramme 1** permet de diminuer le nombre de ces instructions :

```
10 X = X + 1
20 PRINT X, X * X, X* X *
              X, SQR (X)
30 IF X < 10 THEN 10
40 END
```

Après initialisation, X prend la valeur zéro. Lors de la première exécution de la ligne 10 : X = 1. Cette valeur sert aux 1ers calculs, effectués dans l'ordre d'impression PRINT de la ligne 20. Tant que X reste inférieur à 10, il y a retour à la ligne 10 (boucle) ; incrémentation de X (+ 1), nouvelle impression et un nouveau test. Si X = 10, le travail est fini et il n'est pas nécessaire de retourner de nouveau à la ligne 10 (X vaudrait 11 après l'exécution de cette ligne). Ce détournement ou sortie de la boucle détermine l'exécution de l'instruction 40 qui est l'ordre FIN.

Pour simplifier davantage l'écriture des boucles, compte tenu de leur importance on a simplifié leur programmation à l'aide des deux instructions spécialisées FOR et NEXT, ce qui donne :

```
10 FOR X = 1 TO 10
20 PRINT X, X↑2, X↑3,
                 SQR (X)
30 NEXT X
40 END
```

A la ligne 10, X prendra toutes les valeurs entre 1 et 10. La raison de la progression est initialisée à 1 (STEP 1) lorsqu'elle n'est pas spécifiée. Un test comme celui de la ligne 30 de l'exemple précédent est prévu dans le cadre de l'instruction NEXT.

A la ligne 30 on demande de recommencer l'exécution de la boucle avec une nouvelle valeur de X. Cette dernière étant la suivante de celles qu'il peut prendre à la ligne 10.

BOUCLES IMBRIQUÉES

Lorsque l'on est amené à exécuter des boucles à l'intérieur d'autres boucles, certaines précautions s'imposent afin d'éviter de tourner en rond, sans fin, ou de perdre des paramètres.

Une première règle, élémentaire, est d'affecter à chaque boucle une variable distincte. D'autre part, il faut éviter le chevauchement des boucles, comme le montrent les exemples suivants :

Boucles autorisées :

```
┌─FOR I
│┌FOR J
│└NEXT J
└─NEXT I
```

ou bien :

```
┌─FOR I
│┌FOR J
││┌FOR K
││└NEXT K
││┌FOR L
││└NEXT L
│└NEXT J
└─NEXT I
```

Boucles non autorisées :

```
┌FOR I
│┌FOR J
└┼NEXT I
 └NEXT J
```

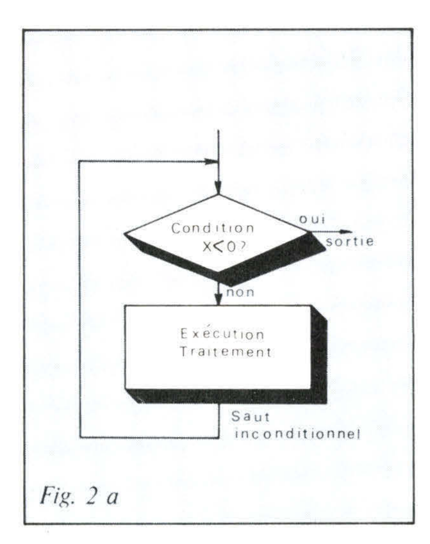


Fig. 2 a

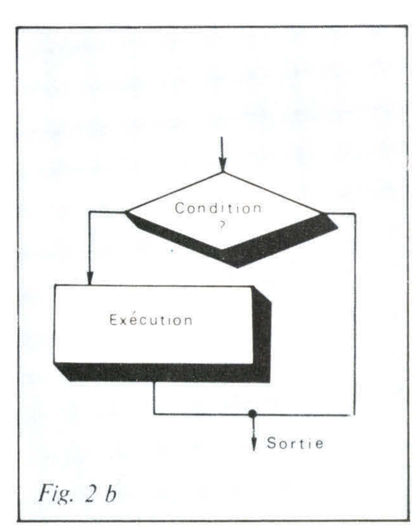


Fig. 2 b

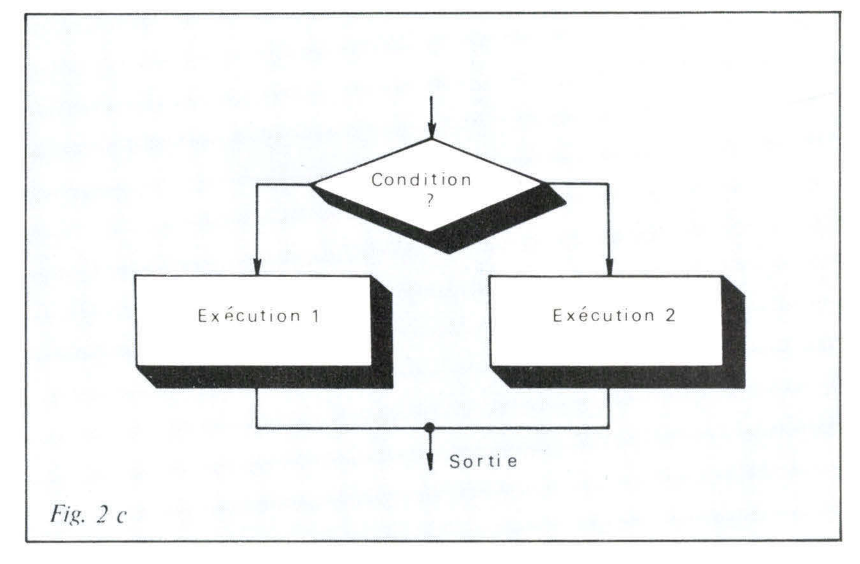


Fig. 2 c

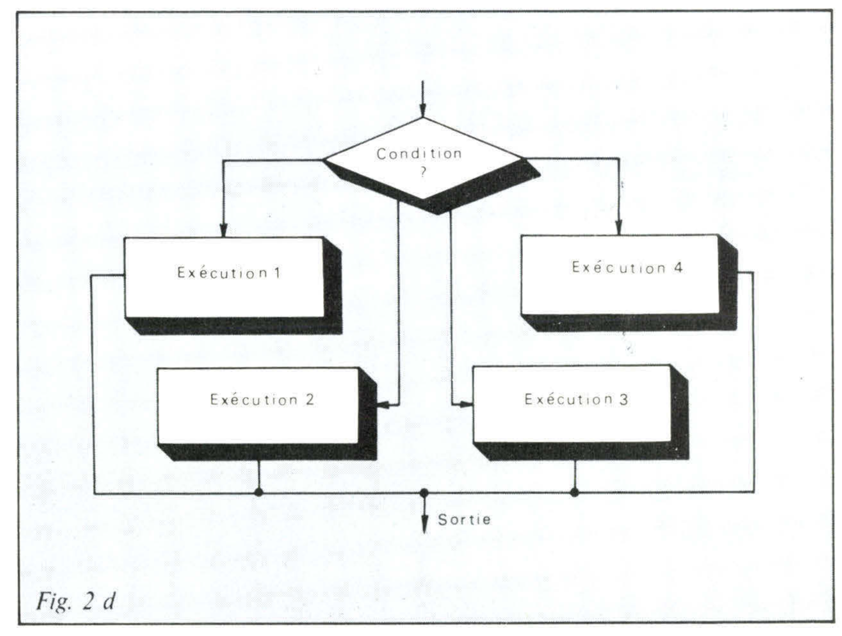


Fig. 2 d

Fig. 2. – Organigrammes des différents types de sauts imaginables.

Les instructions d'un programme sont exécutées normalement dans l'ordre de leur numérotation.

Les SAUTS

Les instructions d'un programme sont exécutées normalement dans l'ordre de leur numérotation. Lorsqu'on a besoin de faire exception à cette règle on utilise une instruction de saut.

Si le saut est impératif, on a recourt à une instruction de saut inconditionnel :

```
100 GO TO 30
```

Si le saut dépend du résultat d'un test il s'agira d'un saut conditionnel. Le test lui-même est exprimé par une instruction IF :

```
100 IF X < = 3 GO TO 30
```

Il y a plusieurs types de sauts, exprimés sur les organigrammes de la **figure 2.** Le programme correspondant à l'organigramme de la **figure 2 a** est le suivant :

```
10 INPUT X
20 IF X < 0 GO TO 50
30 S = S + X
40 GO TO 10
50 PRINT S
60 END
```

Ce programme additionne les nombres X que l'on fait entrer par l'instruction INPUT jusqu'à la rencontre d'un nombre négatif, ce qu arrête le traitement et fait imprimer la somme. La condition est à la ligne 20.

Aux **figures 2 b, 2 c** et **2 d** correspondent des configurations courantes de sauts. Une condition peut aboutir à une, deux ou plusieurs exécutions et traitements ou à un saut vers la sortie sans aucune exécution. ■

Afin de fixer quelque peu nos connaissances en BASIC, voici quelques questions récapitulatives :

I Combien y a-t-il de noms de variables en BASIC ?

II Que signifie l'expression arithmétique A * B / C * D ?

III Une étiquette (numéro de ligne) peut elle être négative ?

IV Signaler la faute contenue dans chacune des expressions suivantes (une seule erreur par expression)
a) B + C = A
b) Y = X ↑ 2 + 2 N
c) Z = (A ↑ 1 . 5) + Y ↑ 2) ↑ 0,5

V Quelle différence y a-t-il entre les instructions ?
40 LET T = 12
40 T = 12

VI Que se passe-t-il si une instruction DATA contient plus de données que n'en demande une instruction READ ?

VII Si l'on suppose que le contenu d'un mot mémoire d'adresse $FFFF_H$ (65535) est FE_H (254), quel sera le résultat de l'exécution de l'instruction :
PRINT PEEK (65535)

I – Combien y a-t-il de noms de variables en BASIC ?

286 — La raison est simple si l'on se rappelle qu'une variable, qui peut être simple ou indicée (tableau), est constituée d'une lettre de l'alphabet (26 en tout), suivie ou non d'un chiffre. Il y a 10 chiffres, 0, 1, 2... 9 et une possibilité sans chiffre. Donc le nombre total de lettres de l'alphabet sera multiplié par 10 + 1 = 11. D'où 26 x 11 = 286 possibilités.

Rappelons à cette occasion que tous les signes de l'ALPHABET du BASIC ne rentrent pas dans la constitution des noms de variables. Mise à part l'ensemble de chiffres et de lettres que l'on désigne sous le nom de CARACTÈRES ALPHANUMÉRIQUES, nous disposons d'une série de CARACTÈRES SPÉCIAUX :

$ ↑ + – * / . , = ( ) blanc

ceux-ci ne rentrent pas dans la composition des noms des variables. Toutes les lettres sont en majuscules.

La distinction entre O et zéro s'obtient en barrant la lettre O : Ø

Les noms des variables indicées ou tableaux respectent les mêmes règles, quel que soit le nombre d'indices. Par exemple, soient deux tableaux, V (I) une liste de vitesses et V (X, Y, Z) une liste de vecteurs, définis chacun par ses coordonnées dans l'espace, dans un même programme, seront confondus et un message d'erreur apparaîtra dans le meilleur des cas.

Rappelons également que si, dans les grandes machines on n'a pas à s'occuper de l'implantation en mémoire des tableaux, dans les petits systèmes un DIM A (50) réserve 50 emplacements de quelques mots-mémoire chacun, pour permettre de loger des éléments A (I) à 9 décimales de précision. A deux ou trois dimensions on arrive très vite à saturation des machines ne connaissant pas l'allocation dynamique de mémoire (allocation de place mémoire au moment où l'on en a besoin).

II – Que signifie l'expression arithmétique A * B/C * D ?

$$\frac{A\,B\,D}{C}$$

Lorsque dans une expression arithmétique il y a plusieurs opérations entre les constantes ou variables, il existe un ordre de priorité entre les

opérations, le même que celui de l'arithmétique de tous les jours :

(parenthèses) ; ↑; * et / ; ±

A priorité égale, toutes les opérations s'effectuent de gauche à droite. Dans notre cas, la multiplication et la division étant des opérations d'égale priorité, l'ordre des calculs, de gauche à droite, sera A * B ; résultat divisé par C ; résultat multiplié par D. C'est peut-être la seule entorse à la règle qui consiste à exprimer le plus naturellement possible les opérations dont on demande l'exécution entorse, liée à la nécessité graphique de poser les formules sur une seule ligne. De la même façon, les divers types de parenthèses, rondes, carrées, accolades, etc., utilisées dans l'écriture courante sont remplacées par la répétition autant de fois que nécessaire du même type de parenthèse :

[(a + b) * c + e * d] s'écrit ((a + b) * c + e * d)

Pour finir, si l'on voulait écrire l'expression

$$\frac{A\,B}{C\,D}$$

en BASIC, on obtiendrait, sur une seule ligne :

A * B / (C * D).

III – Une étiquette (numéro de ligne) peut-elle être négative ?

Non on utilise toujours un nombre positif et sans signe (naturel), de 4 ou 5 chiffres. Sur les machines évoluées on peut rencontrer des nombres fractionnaires, permettant l'insertion facile de 10, 100 ou 1000 instructions entre deux lignes distantes d'une unité :

1
. . . 1.1 ; 1.01 ; 1.001
. . . 1.9 ; 1.99 ; 1.999
2 . . . . .

Si l'on veut insérer une ligne, avant celle dont le numéro est 0, on peut passer par une rénumérotation des lignes de 10 en 10, par exemple, à l'aide d'une directive NUM 1 ou NUM 10, existante sur le plupart des systèmes.

IV – Signaler la faute contenue dans chacune des expressions suivantes (il y a une seule erreur par expression) :

a) B + C = A
b) Y = X↑2 + 2 N
c) Z = (A↑1.5) + Y↑2)↑0.5

a) L'expression arithmétique doit figurer à droite du signal égal et la variable A doit se trouver à gauche, donc inverser l'égalité.

b) Cette erreur se produit si souvent qu'il est bon de la souligner : 2N en écriture courante signifie une multiplication implicite de 2 par N. La machine ne la comprend pas et ne doit pas la comprendre ainsi, car dans le cas contraire, un nom de variable, A5 par exemple, pourrait être confondu avec la multiplication de A par 5. Donc, écrire 2 * N à la place de 2N.

c) Le compte du nombre de parenthèses n'est pas exact. Une règle simple que respecte tout détecteur d'erreurs de syntaxe dans chaque interpéteur BASIC, est de faire la somme des parenthèses à droite et à gauche et de s'assurer de la stricte égalité. Ici on trouve une parenthèse à droite en trop. On doit donc écrire, correctement :

Z = ((A↑1.5) + Y↑2)↑0.5

V – Quelle différence y a-t-il entre les instructions :

40 LET T = 12
40 T = 12

Aucune. On utilise le mot clé « LET » pour donner une forme plus littérale à un programme. Il définit une affectation qui peut être implicite (dans la deuxième ligne).

VI – Que se passe-t-il si une instruction DATA contient plus de données que n'en demande une instruction READ ?

Exemple :

```
100 READ A, B, C
300 DATA 1, 2, 3, 4, 5, 6, 7, 8, 9
310 END
```

Rien de grave. La machine exécute néanmoins son programme. Bien souvent on utilise même une boucle qui produit l'exécution du READ plusieurs fois de suite. Dans notre exemple, trois passages par la ligne 100 produiront l'entrée de toutes les données (il y en a 9, lues par 3 à la fois).

Le cas contraire, celui d'un nombre de données insuffisantes produit un message d'erreur.

VII – Si l'on suppose que le contenu d'un mot-mémoire d'adresse $FFFF_H$ (65535) est FE_H (254), quel sera le résultat de l'exécution de l'instruction :

PRINT PEEK (65535)

L'impression de 254 en décimal. La manipulation d'octets utilise la notation décimale, sauf spécifications contraires.

Programme financier

```
10 REM CE PROGRAMME FABRIQUE UN PLAN DE FINANCEMENT
20 REM ON DEMANDE A LA BANQUE UN CAPITAL DE "C"FRS
30 REM A UN TAUX DE X% L'AN, A ECHEANCES CONSTANTES
40 REM SUR "N" ANNEES
45 PRINT ON
50 INPUT "CAPITAL PRETE",C
52 INPUT "DUREE DU PRET",N
54 INPUT "ANNEE",I
60 INPUT "TAUX D'INTERET",X
62 B=C
63 PRINT
70 A=C*X/(1-(1/((1+X)^N)))
71   PRINT " CAPITAL       ANNEE          ANNUITES   AMORTISSEMENT      VALEUR RESIDUELLE"
72 FOR N1=1 TO N STEP 1
75 R=B*X
76 S=A-R
77 F=N1+I
80 B=B-S
90 PRINT C,F,A,S,B
95 NEXT N1
98 PRINT OFF
100 END
```

Ce programme permet d'établir un plan de financement. Un capital de « C » francs emprunté à « X » % l'an et à annuités fixes sur « N » années, s'amortit selon un tableau dont on spécifie par colonne : le capital de départ, l'année en cours, les annuités (fixes), l'amortissement et la valeur résiduelle.

Jusqu'ici un tel plan était fourni par les grands organismes de crédit disposant d'ordinateurs puissants. A présent, l'informatique individuelle permet à tout un chacun de calculer de façon précise le financement de tel ou tel emprunt et vous donnera la possibilité d'obtenir aisément le même listing que celui délivré par les banques.

Ce programme présente quelques particularités

Il « tourne » sur une machine ZILOG, conçue initialement en tant que système de mise au point microprocesseur, mais dotée finalement du BASIC.

Le taux d'intérêt X s'exprime d'une façon décimale : 10 % s'écrira 0,1.

L'annuité fixe d'un tel emprunt se calcule selon la formule (voir la démonstration dans l'encadré) :

$$A = \frac{X}{1 - (1 + X)^{-N}} \cdot C$$

La colonne des années affiche les années **à venir,** à partir de l'année en cours, « I ».

Description détaillée du programme

10 à 40	Commentaires sans action sur la machine.
45	PRINT ON signifie sur le système ZILOG une validation sur l'imprimante. En l'absence de cette directive les résultats apparaissent uniquement sur l'écran cathodique.
50,52,54,60	Entrée des données du plan : capital prêté, « C » ; durée du prêt en années, « N » ; année en cours, « I » ; taux d'intérêt, « X ». INPUT est une entrée en mode conversationnel. Lors de l'exécution du programme l'ordinateur demande tour à tour le capital prêté, la durée du prêt, l'année en cours et le taux d'intérêt par un point d'interrogation.
62	B représente la valeur résiduelle, c'est-à-dire ce que l'on devrait verser si l'on voulait rembourser l'intégralité du prêt à un certain moment. Initialement B = C, le capital prêté.
63	Un « PRINT » vide signifie une ligne vide ou saut d'une ligne lors de l'impression du listing.
70	Calcul des annuités une fois pour toute d'après la relation donnée ci-dessus.
71	Impression de l'intitulé.
72 à 95	Boucle d'impression des résultats des calculs pour les « N » années. L'amortissement « S » se calcule selon la formule : $S = A - B . X$

77 La variable F représente l'année courante. Elle s'obtient en additionnant l'année du prêt à l'année en cours (ici 1979).

80 Calcul de la nouvelle valeur résiduelle.

90 Impression des résultats pour chaque année de calcul.

98 PRINT OFF déconnecte l'imprimante dans le système ZILOG.

L'exemple traité concerne le remboursement à 10 % l'an (X = 0,1 et non pas 10...) d'un capital de 10 000 F prêté sur 20 ans. Pour un château en Savoie, multipliez le capital par 100... ■

André DORIS

Calcul des annuités (ou mensualités) fixes de remboursement d'un capital C emprunté à un taux de X % sur N années

Nous pouvons évaluer dans un premier temps la valeur résiduelle (B_1), c'est-à-dire la somme restant à rembourser après la première année et après le paiement de la première annuité (A).

$$B_1 = C\,(1 + X) - A$$

En effet, le capital a pris de la valeur et est devenu C(1 + X) mais A francs ont été versés.

La valeur résiduelle au bout de la 2e année de remboursement

$$B_2 = [C\,(1 + X) - A]\,(1 + X) - A$$

La somme restant due était C (1 + X) – A à la fin de la seconde année, cette somme a donc été multipliée par le facteur (1 + X). B_2 peut aussi s'exprimer sous la forme :

$$B_2 = C\,(1 + X)^2 - A\,(1 + X) - A$$

De la même façon la valeur résiduelle de la 3e année sera :

$$B_3 = [C\,(1 + X)^2 - A\,(1 + X) - A]\,(1 + X) - A$$

ou

$$B_3 = C\,(1 + X)^3 - A\,(1 + X)^2 - A\,(1 + X) - A$$

Ainsi nous pouvons écrire que la valeur résiduelle à l'année N est :

$$B_N = C\,(1 + X)^N - A\,(1 + X)^{N-1} - A\,(1 + X)^{N-2} - \dots - A$$

Or B_N doit être nulle car au bout de la $N^{ième}$ année, l'intégralité du prêt est remboursée par conséquent :

$$B_N = 0 = C\,(1 + X)^N - A\,(1 + X)^{N-1} - A\,(1 + X)^{N-2} - \dots - A$$

Nous sommes donc en mesure d'évaluer A ; le montant des annuités à rembourser chaque année :

$$A\,(1 + X)^{N-1} + A\,(1 + X)^{N-2} \dots + A = C\,(1 + X)^N$$

et

$$\boxed{A = C \cdot \frac{(1 + X)^N}{(1 + X)^{N-1} + (1 + X)^{N-2} + (1 + X)^{N-3} \dots + 1}}$$

Nous pouvons simplifier le calcul de cette expression en remarquant que :

$$(1 + X)^N = X\,[(1 + X)^{N-1} + (1 + X)^{N-2} + \dots + 1] + 1$$

d'où :

$$(1 + X)^{N-1} + (1 + X)^{N-2} + \dots + 1 = \frac{(1 + X)^N - 1}{X}$$

Si nous reportons cette relation dans l'expression de A :

$$A = C \cdot \frac{(1 + X)^N}{\dfrac{(1 + X)^N - 1}{X}}$$

$$A = C \cdot \frac{X\,(1 + X)^N}{(1 + X)^N - 1}$$

En divisant la fraction haut et bas par $(1 + X)^N$

$$A = C \cdot \frac{X}{1 - \dfrac{1}{(1 + X)^N}}$$

et

$$\boxed{A = C \cdot \frac{X}{1 - (1 + X)^{-N}}}$$

```
>CAPITAL PRETE?  10000
DUREE DU PRET?  20
ANNEE?   1978
TAUX D'INTERET?  .1

 CAPITAL        ANNEE         ANNUITES    AMORTISSEMENT     VALEUR RESIDUELLE
 10000          1979          1174.59        174.596          9825.4
 10000          1980          1174.59        192.056          9633.34
 10000          1981          1174.59        211.262          9422.09
 10000          1982          1174.59        232.388          9189.7
 10000          1983          1174.59        255.627          8934.07
 10000          1984          1174.59        281.189          8652.88
 10000          1985          1174.59        309.308          8343.57
 10000          1986          1174.59        340.239          8003.33
 10000          1987          1174.59        374.263          7629.07
 10000          1988          1174.59        411.689          7217.38
 10000          1989          1174.59        452.858          6764.52
 10000          1990          1174.59        498.144          6266.37
 10000          1991          1174.59        547.959          5718.42
 10000          1992          1174.59        602.755          5115.66
 10000          1993          1174.59        663.03           4452.63
 10000          1994          1174.59        729.333          3723.3
 10000          1995          1174.59        802.266          2921.03
 10000          1996          1174.59        882.493          2038.54
 10000          1997          1174.59        970.742          1067.79
 10000          1998          1174.59        1067.81         -.017944
```

Réalisez votre micro-ordinateur « Micro-Systèmes 1 »

C'est avec enthousiasme que nous repartons avec vous dans ce numéro 3 pour l'étude de « Micro-Systèmes 1 ».

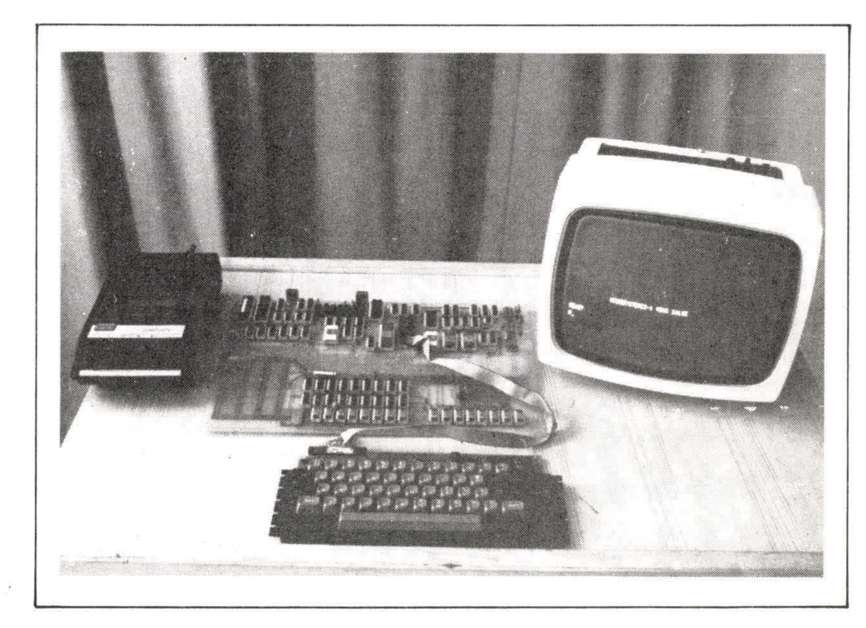
Il est certain que cette idée a fait, parmi nos lecteurs, de nombreux adeptes qui n'ont pas manqué de nous manifester leur intérêt.

Bien que vous ayez de bonnes raisons de vouloir passer rapidement à la réalisation de ce micro-ordinateur, vous ne devez pas perdre de vue qu'il s'agit avant toute chose d'en réaliser en commun l'étude, après quoi nous pourrons passer à son montage.

C'est la raison pour laquelle nous souhaiterions suivre avec vous un plan de travail qui nous permettra de ne pas brûler les étapes et qui, de ce fait, vous garantira le maximum de réussite.

Nous pouvons dire que, pour le moment, nous construisons un corps sain (hardware) dans lequel nous mettrons un peu plus tard un esprit sain (software).

Mais attention, les deux sont liés.



Un circuit imprimé de 42 x 27,5 cm supporte les quelques 88 circuits intégrés de « Micro-Systèmes 1 ».

Avant même de repartir ensemble dans la découverte de cette « étude personnalisée » qu'est « Micro-Systèmes 1 », regardons un instant comment se décompose la présentation de celle-ci.

A – Dans notre numéro précédent nous avons d'une part, exposé les **raisons de cette étude** et d'autre part vous avez pris connaissance de la **fiche technique** de ce micro-ordinateur.

B – A présent, il s'agit d'aborder la description de son **organisation générale** à laquelle nous ajouterons une **analyse du matériel.**

C – Une fois l'étude théorique menée à son terme, il sera temps de passer à la **réalisation pratique** de « Micro-Systèmes 1 ».

D – Il est logique après ceci de consacrer une large part à **l'explication de son fonctionnement.** En outre, cette partie sera l'occasion pour nous d'apporter les **réponses à vos questions** qui sont déjà nombreuses.

E – Parvenus à ce stade, vous êtes en possession d'un véritable micro-ordinateur avec un Basic de 8 K ROM et une mémoire RAM de 2 x 16 K octets. Nous traiterons ensemble des **exemples d'applications** représentatifs des divers domaines d'activités dans lesquels votre système est de taille à apporter une aide précieuse.

Sur le plan technique, le premier contact, que nous avons eu ensemble avec « Micro-Systèmes 1 » reposait essentiellement sur la présentation de sa fiche technique et sur la description de son organisation générale, résumée dans le synoptique de la **figure 1 *.**

Fig. 1. – N° 2. Micro-Systèmes, p. 69.

Nous allons à présent aborder la partie qui, sans nul doute, est la plus délicate de toute votre étude : il s'agit de l'analyse détaillée de chaque élément constituant votre micro-ordinateur.

Autant vous prévenir tout de suite, certains d'entre vous éprouverons quelques difficultés dans cette approche. Que cela ne les décourage en aucune façon. Vous avez deux mois pour travailler cette partie et, en outre, nous sommes là pour réaliser cette étude ensemble et il vous appartient de nous faire part de vos problèmes à ce sujet en nous écrivant. Il vous sera répondu dans nos prochains numéros.

Ceci dit, cette analyse est nécessaire. C'est en connaissant son matériel jusque dans sa plus profonde intimité qu'il vous est permis de vivre en parfaite harmonie avec lui et c'est ce que vous recherchez (il en va d'un matériel comme des hommes...).

Mais, venons-en à cette deuxième partie qu'est ce **chapitre B :**

Organisation générale et analyse du matériel

Nous avons choisi de commencer la description de « Micro-Systèmes 1 » par la partie mémoire. C'est elle en effet qui distingue notre ordinateur de tout autre système à microprocesseur (c'est la mémoire qui fait l'ordinateur).

En effet, si l'on considère des appareils à microprocesseur comme les alarmes, les jeux TV, etc., ces systèmes ont certainement de grands programmes de fonctionnement, mais ils sont figés en mémoire morte ou ROM. On ne trouve nulle part ailleurs 16 K et même 32 K octets de mémoire vive (RAM) prête à recevoir le programme utilisateur. C'est cela le propre d'un ordinateur.

Un commutateur électronique place successivement sur le bus d'adresse la partie haute, la partie basse de l'adresse et les adresses à rafraîchir.

Schéma général de fonctionnement

Le schéma général de fonctionnement est représenté sur le dépliant **(fig. 2)**. Pour mieux l'analyser, nous allons le partager en plusieurs parties :

B.1. La mémoire RAM dynamique de 32 K octets et ses circuits de rafraîchissement, comprenant les boîtiers M_1 à M_{16} et tous les circuits de gestion ; B_4, B_5, B_9, B_{11}, B_{12}, B_{13}, B_{14} et B_{15}.

B.2. Le contrôleur de DMA* et l'unité centrale 6800, les deux principaux animateurs du schéma, tamponnés par B_6, B_7, B_8 pour les adresses et B_0, B_1 pour les données.

B.3. La mémoire de programme M_0 et les décodages d'adresse permettant de lui substituer l'AMORCE du DOS *, nécessaire au fonctionnement des disques souples.

B.4. L'interface minicassette comprenant l'ACIA * U_4 et tous les circuits nécessaires à l'émission/réception de signaux sinusoïdaux à partir des signaux carrés de l'ACIA.

B.5. L'interface vidéo, constitué du contrôleur CRT * U_3, de la PROM *-fusibles M_{24}, de la mémoire d'écran 1 k x 7 bits (M_{17} à M_{23}), du générateur de caractères M_{25} muni des mémoires d'entrée U_{31}, U_{32}, du compteur de points/caractère U_{34} et du registre à décalage U_{33}.

B.6. La PIA * de dialogue clavier encodé 8 bits parallèle-écran de visualisation, U_2 et l'ACIA de liaison série TTY*, RS 232*, boucle de courant 20 mA.

Non figurés sur ce schéma, mais ayant une égale importance :

- Le contrôleur de floppy disk.
- Le contrôleur de Bus GPIB *.
- PIA-S et ACIA-S en option.

Ces derniers constituants prennent place à la verticale de la plaque-mère, sur des supports de bus.

L'agent de liaison entre ces parties est un boîtier contenant des amplificateurs de ligne. Nous allons donc commencer notre étude par l'analyse du fonctionnement de ces boîtiers qui portent une notation en « B » (de « **Buf**fer » en anglais).

Le schéma utilise trois sortes de « buffers » ou « tampons », les 8T26, 8T95 et 8T97.

L'organisation interne de ces boîtiers est celle de la **figure 3.**

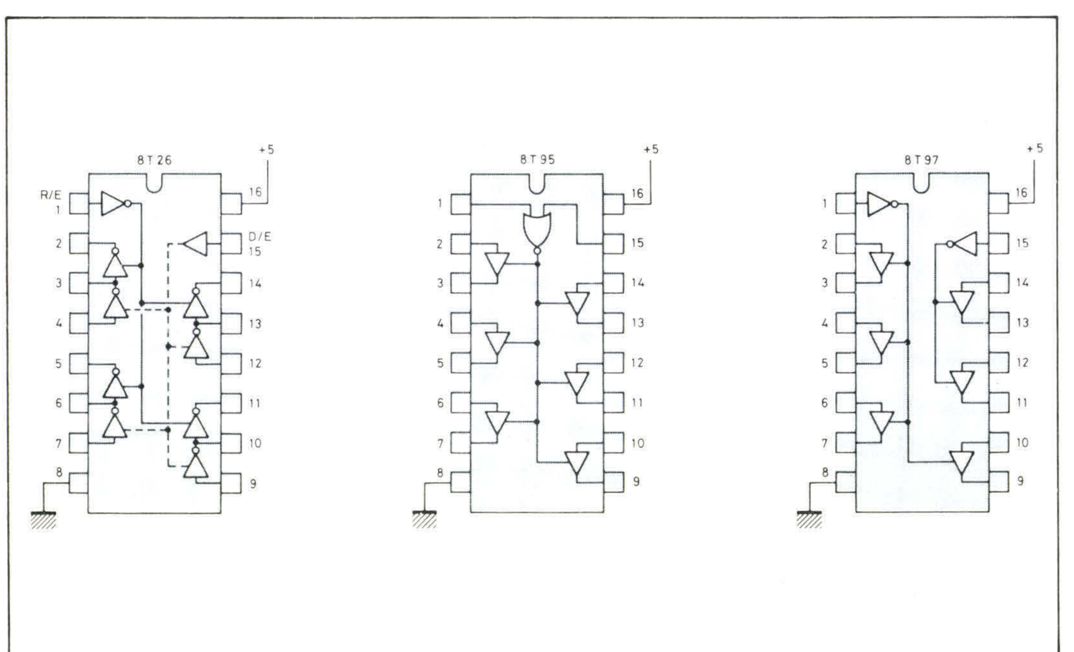
Nous constatons la présence d'amplificateurs, inverseurs ou non, qui reçoivent sur le flanc une commande de mise à l'état haute-impédance de leur sortie. Par convention, un « 1 » logique appliqué sur cette entrée de commande rend les sorties actives et un « 0 » les met à l'état haute impédance (ou tri-state), auquel cas le circuit coupé se manifeste en sortie par un faible courant de fuite.

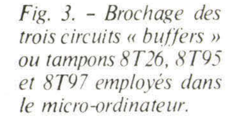
Ces trois boîtiers peuvent être classés en deux catégories :

Le 8T 26 est un « aller-retour », utilisé habituellement sous forme de quatre paires d'amplificateurs tête-bêche, formant un amplificateur bidirectionnel de 4 fils.

Les 8T 95, 8T 97 sont des « aller simple », ne différant que par la commande de mise en « tristate » de leurs amplificateurs : le 8T 97 possède deux commandes, pouvant se séparer en deux groupes de deux et de quatre amplificateurs, alors que le 8T 95 répond avec les 6 amplificateurs simultanément, ses sorties se mettant à l'état haute-impédance dès qu'un « 1 » logique apparaît sur l'une ou l'autre des entrées du NOR (OU logique et inversion).

Les exemples d'applications du 8T 26 de la **figure 3a** montrent




Fig. 3. - Brochage des trois circuits « buffers » ou tampons 8T26, 8T95 et 8T97 employés dans le micro-ordinateur.


** DMA : Direct Memory Acces (accès direct à la mémoire).*

** DOS : Disk Operating System (système de contrôle et de gestion pour floppy).*

** ACIA : Asynchronous Communications Interface Adapter (boîtier facilitant les communications série).*

** CRT : Cathode Ray Tube ou tube à rayons cathodique (écran TV).*

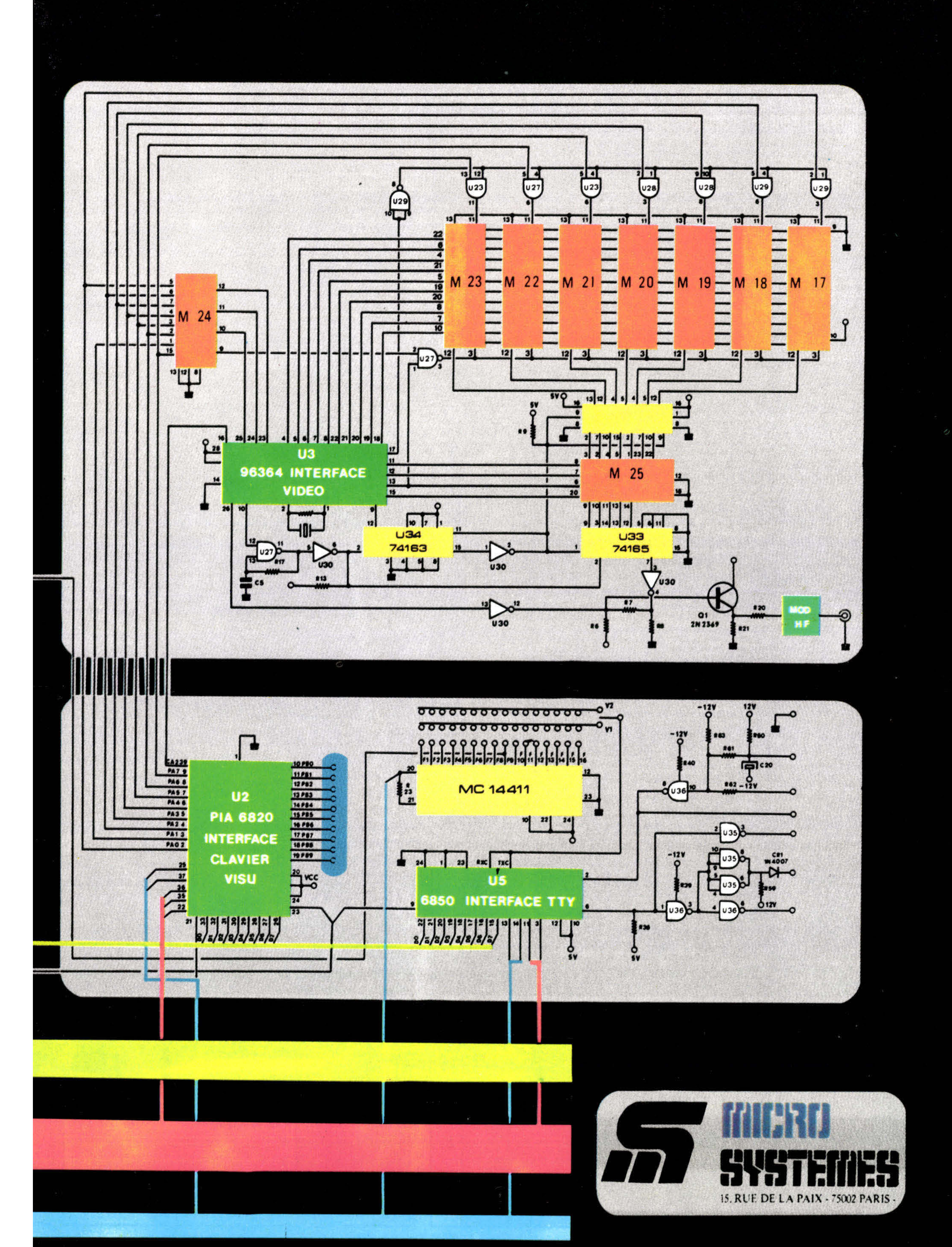
** PROM : Programmable Read Only Memory (mémoire morte programmable).*

** PIA : Peripheral Interface Adapter (adaptateur d'interface pour périphérique).*

** TTY : Teletype writter (équipement télétype).*

** RS232 : Spécifications de transmission série (190, 300, 1200, 2400, 4800 et 9600 bauds).*

** GPIB : General Peripheral Interface Bus (Bus d'instrumentation, réalise la liaison entre 16 instruments et leur permet de transmettre des données, résultats...).*



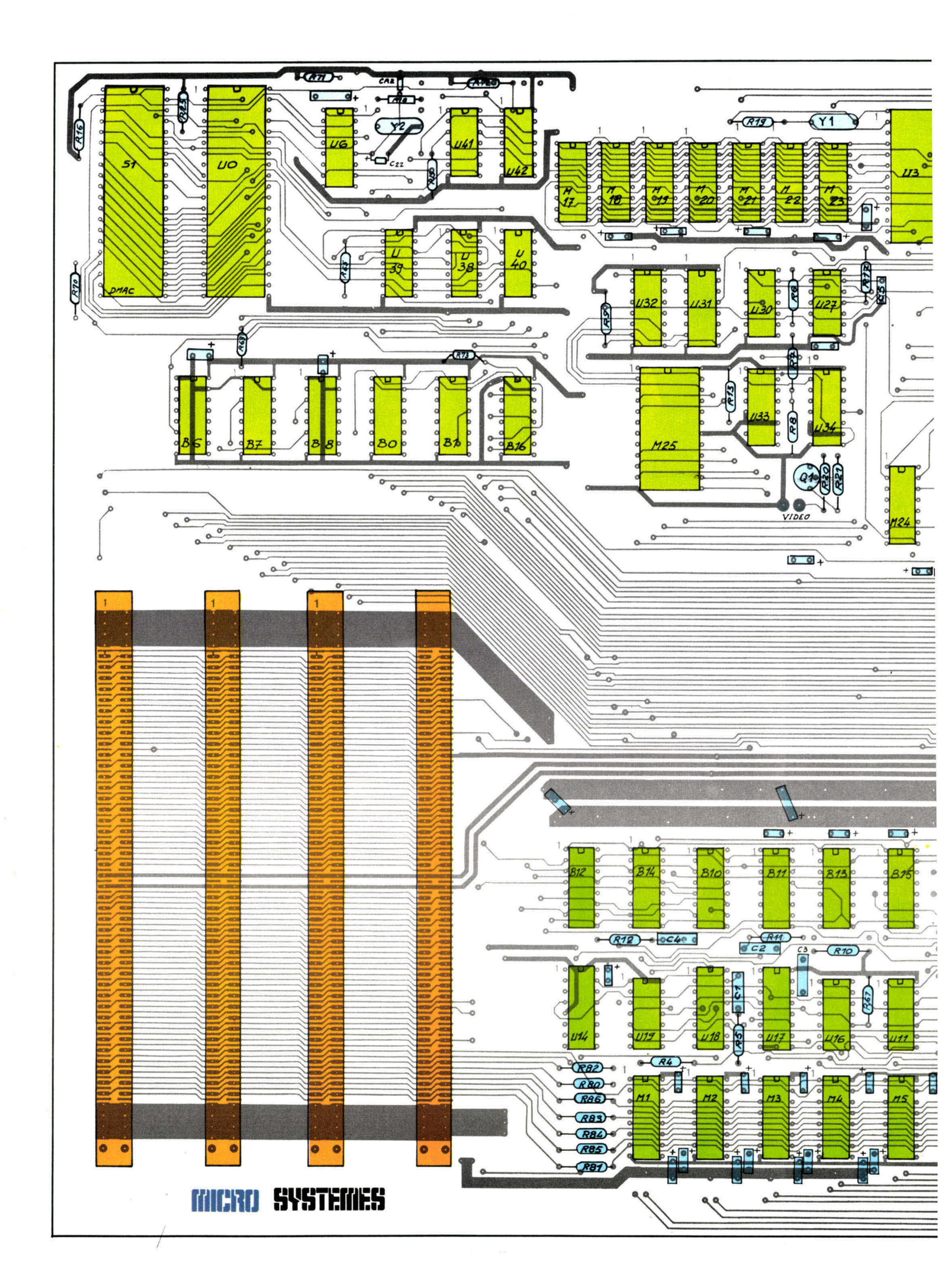
M 24
M 23
M 22
M 21
M 20
M 19
M 18
M 17
U23
U27
U28
U29
U3
96364 INTERFACE
VIDEO
M 25
U34
74163
U33
74165
U30
Q1
2N 2369
MOD
HF
U2
PIA 6820
INTERFACE
CLAVIER
VISU
VCC
MC 14411
U5
6850 INTERFACE TTY
U35
U36
-12V
12V
5V
MICRO
SYSTEMES
15, RUE DE LA PAIX - 75002 PARIS -

NOMENCLATURE « MICRO-SYSTEMES 1 »

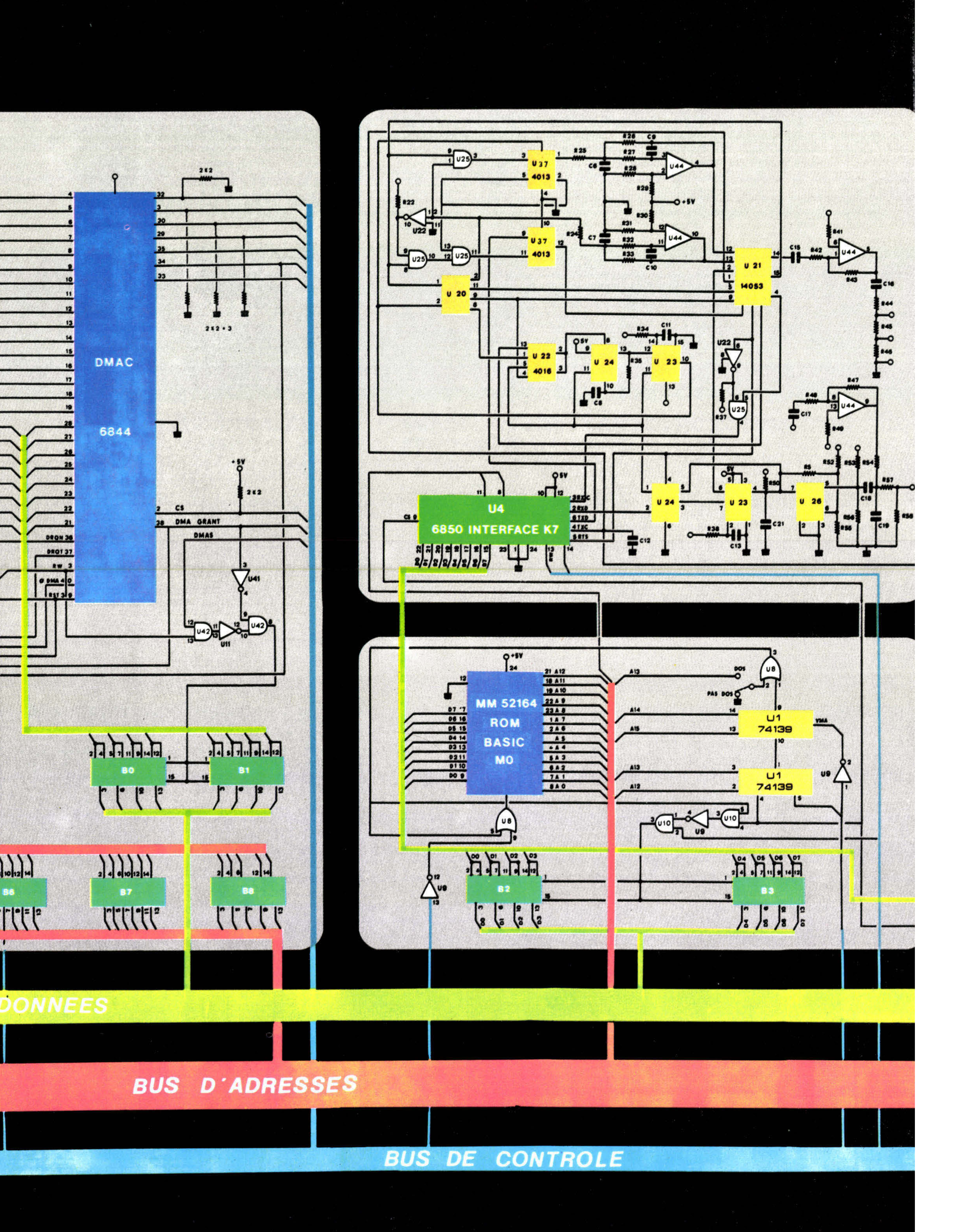
NOMENCLATURE

CIRCUITS LOGIQUES DE GESTION			
Ref.constructeur	**Ref. Micro-Systèmes**	**Désignation**	**Qté**
MC 6800	U_0	Unité Centrale	1
74 LS 139	U_1	Décodeur-Démultiplexeur	1
MC 6820	U_2	PIA (clavier-visu)	1
96364	U_3	Interface vidéo	1
6850	U_4 ; U_5	ACIA (TTY K7)	2
96875	U_6	Horloge	1
74 LS 10	U_7	Portes « NAND » à 3 entrées	1
74 LS 32	U_8 ; U_{40}	Portes « OU »	2
74 LS 04	U_9 ; U_{12} ; U_{41}	Inverseurs	3
74 LS 08	U_{10} ; U_{28} ; U_{29} ; U_{39} U_{42} ;	Portes « ET » à 2 entrées	5
74 LS 11	U_{11}	Porte « ET » à 3 entrées	1
74 LS 00	U_{13} ; U_{38}	Porte « NAND » à 2 entrées	2
74 LS 75	U_{14} ; U_{15}	Quadruple « Latch »	2
74 LS 74	U_{16}	Double bascule D	1
8602	U_{17} ; U_{18}	Double monostable	2
4024	U_{19} ; U_{20}	Compteur binaire	2
4053	U_{21}	Multiplexeur analogique	1
4016	U_{22}	Quadruple « switch » analogique	1
14538	U_{23}	Double monostable	1
4013	U_{24}	Bascule D	1
4081	U_{25}	Quadruple « NAND » à 2 entrées	1
75140	U_{26}	Double « Receiver » de ligne	1
74 S 132	U_{27}	Quadruple « NAND » à 2 entrées analogiques	1
74 LS 05	U_{30}	Sextuple inverseur à collecteurs ouverts	1
74 LS 175	U_{31} ; U_{32}	Quadruple bascule D	2
74165	U_{33}	Registre à décalage	1
74 LS 163	U_{34}	Compteur synchrone 4 bits	1
1488	U_{35}	Driver TTY	1
1489L	U_{36}	Driver TTY	1
MC14013	U_{37}	Double « Flip-Flop »	1
MC3301	U_{44}	Quadruple Ampli-op.	1
14411	U_{43}	« Bit Rate Generator »	1
MC6844	S_1	Contrôleur d'accès direct à la mémoire	1
SV6726	M'_1 ; M'_2 ; M'_3 ; M'_4 ; M'_5 ; M'_6 ; M'_7 ; M'_8	Boîtiers mémoire 16 K x 1 bit	8
2102-3	M_{17} ; M_{18} ; M_{19} ; M_{20} ; M_{21} ; M_{22} ; M_{23} ; M_{24}	RAM (mémoire d'écran)	7
{743287 DC III (Proteus)	M_{24}	Mémoire caractères	1
{GC III (Proteus) (2708)	M_{25}	Générateur de caractères	1
8T26	B_0 ; B_1 ; B_2 ; B_3 B_4 ; B_5	« Buffers »	6
8T97	B_6 ; B_7 ; B_8 ; B_9 B_{10} ; B_{11} ; B_{16}	« Buffers »	7
8T95	B_{12} ; B_{13} ; B_{14} ; B_{15}	« Buffers »	4

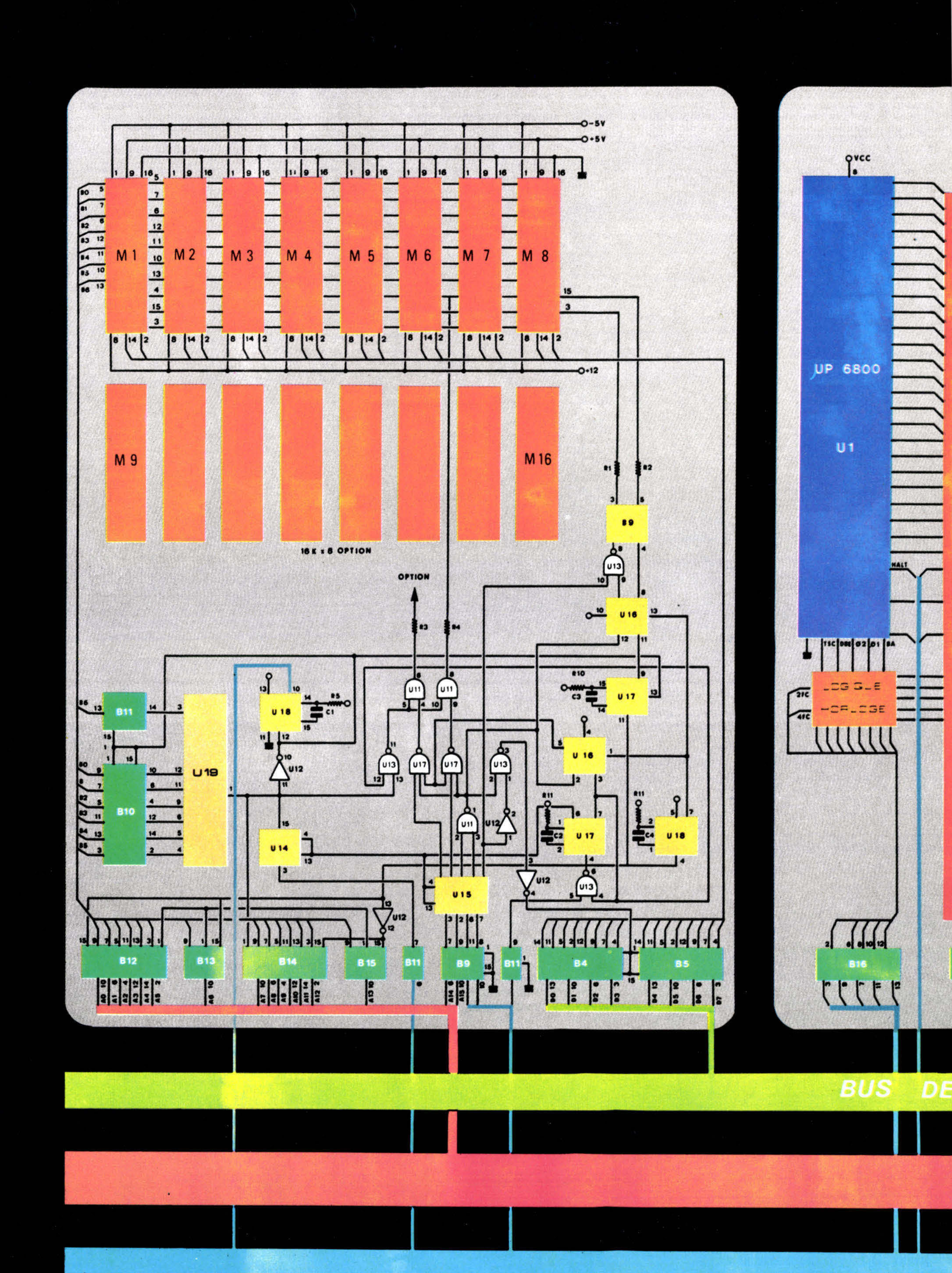
RÉSISTANCES		
Valeurs	**Ref. Micro-Systèmes**	**Qté**
22 Ω	R_1 ; R_2 ; R_3 ; R_4 ; R_{18} ; R_{67} ; R_{68} ; R_{80} ; R_{81} ; R_{82} ; R_{83} ; R_{84} ; R_{85} ; R_{86}	14
8,2 kΩ	R_5	1
330 Ω	R_6	1
220 Ω	R_7	1
2,2 kΩ	R_8 ; R_{25} ; R_{59} ; R_{61} ; R_{63} ; R_{69} ; R_{71} ; R_{72} ; R_{73}	9
1 kΩ	R_9 ; R_{13} ; R_{15} ; R_{45} ; R_{50} ; R_{57} ; R_{66} ; R_{70}	8
5,6 kΩ	R_{10} ; R_{11} ; R_{64}	3
6,2 kΩ	R_{12}	1
10 kΩ	R_{16} ; R_{22} ; R_{34} ; R_{35} ; R_{36} ; R_{37} ; R_{44} ; R_{52} ; R_{53} ; R_{55} ; R_{56} ; R_{60} ; R_{65}	13
470 Ω	R_{17}	1
2 MΩ	R_{19}	1
68 Ω	R_{20}	1
150 Ω	R_{21}	1
1 MΩ	R_{23}	1
100 kΩ	R_{24} ; R_{26} ; R_{28} ; R_{31} ; R_{33} ; R_{38} ; R_{41} ; R_{49}	8
27 kΩ	R_{27} ; R_{32}	2
180 kΩ	R_{29} ; R_{30}	2
68 kΩ	R_{39} ; R_{40}	2
22 kΩ	R_{42} ; R_{43} ; R_{48} ; R_{51}	4
100 Ω	R_{46} ; R_{58}	2
270 kΩ	R_{47}	1
560 Ω	R_{62}	1
	CONDENSATEURS	
10 nF	C_1	1
150 pF	C_2	1
33 pF	C_3	1
22 pF	C_4	1
82 pF	C_5	1
2,2 nF	C_6 ; C_9 ; C_{11}	3
1 nF	C_7 ; C_8 ; C_{10} ; C_{12}	4
2,3 nF - 5 %	C_{13}	1
47 nF	C_{14} ; C_{19}	2
100 nF	C_{15} ; C_{16} ; C_{24} ; C_{25}	4
220 nF	C_{17} ; C_{18}	2
1 μF - 15 V Tantale	Tous les condensateurs de découplage	48
10 μF	C_{22}	1
1 μF - 63 V	C_{23}	1
	TRANSISTORS	
2N 2369	Q_1	1
	DIODES	
1N 4007	CR_1	1
1N 4148	CR_2	1
	QUARTZ	
1,008 MHz	Y_1	1
3,579 MHz	Y_2	1
	REGULATEURS - 5 V	
MC 7905	RG	1

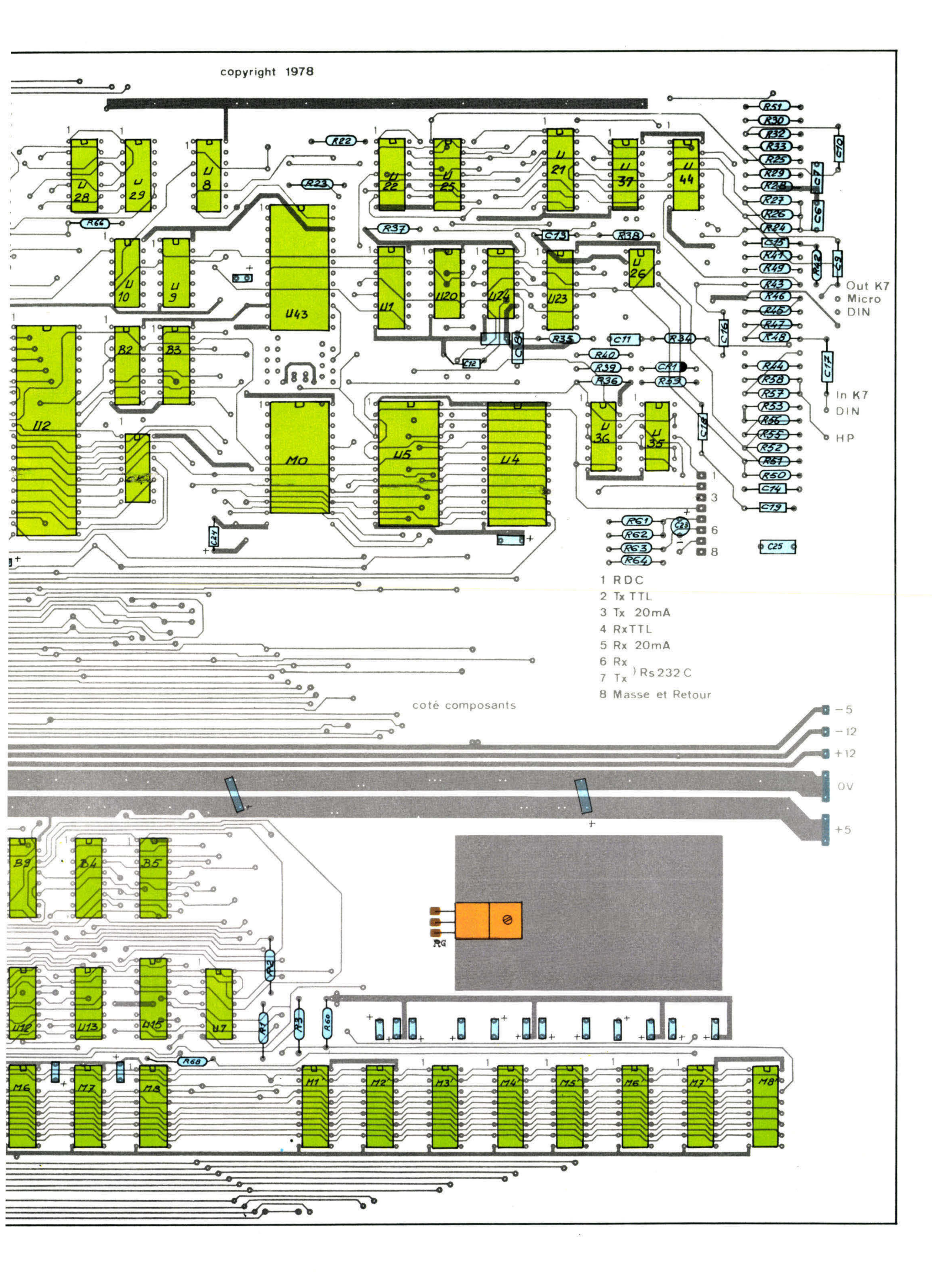


MICRO SYSTEMES
VIDEO



DMAC
6844
U4
6850 INTERFACE K7
MM 52164
ROM
BASIC
MO
U1
74139
U 21
14053
U 37
4013
U 22
4016
B0
B1
B2
B3
B6
B7
B8
DONNEES
BUS D'ADRESSES
BUS DE CONTROLE


-5V
+5V
M 1
M 2
M 3
M 4
M 5
M 6
M 7
M 8
+12
M 9
M 16
16 K x 8 OPTION
OPTION
B9
U13
U16
U17
U11
U18
U19
U12
U14
U15
B10
B11
B12
B13
B14
B15
B4
B5
VCC
UP 6800
U1
HALT
B16
BUS


copyright 1978
Out K7
Micro
DIN
In K7
DIN
HP
1 RDC
2 Tx TTL
3 Tx 20mA
4 RxTTL
5 Rx 20mA
6 Rx
7 Tx) Rs 232 C
8 Masse et Retour
coté composants
−5
−12
+12
0V
+5
RG

d'une part le tamponnage d'un bus de données de 4 fils entre une unité centrale et une mémoire, le sens d'amplification étant déterminé par le signal de lecture/écriture (R/W) du microprocesseur lui-même et, d'autre part, une utilisation dans un circuit de décodage d'adresse.

Dans ce dernier cas il est parfois élégant de pouvoir disposer des fils d'adresse directs et inversés, amplifiés et sélectionnés à l'aide de roues codeuses (les contacts S_1, S_2, S_3, S_4 appartiennent à une roue codeuse hexadécimale). Placer une mémoire, dont le fil de sélection se trouve en aval du boîtier NAND-TTL, à telle ou telle adresse devient très simple, car l'indication binaire de la roue codeuse pourrait représenter en même temps un chiffre de l'adresse-mémoire sélectionnée.

Les 8T 95 et 8T 97 sont utilisés principalement au partage des bus entre plusieurs utilisateurs possibles.

Dans l'exemple de la **figure 3b,** un sélecteur de commande à trois positions (électronique, mais représenté comme un commutateur mécanique) peut placer sur le bus commun, consécutivement, les informations présentes sur les bus 1, 2 et 3.

Dans notre micro-ordinateur nous utilisons cette technique pour l'adressage et le rafraîchissement de nos mémoires dynamiques, les bus 1, 2 et 3 étant reliés respectivement à la partie haute (A_7–A_{13}) et basse (A_0–A_6) du bus d'adresses et à un compteur binaire à 7 étages (U_{19}) générant les adresses à rafraîchir.

Les problèmes liés à l'emploi des tampons de ligne sont multiples. On leur demande, bien entendu, d'amplifier au maximum les signaux et d'être si possible compatibles MOS ou LS (Low Schottky) en entrée. Or, il y a un impératif très important, qui est celui de la vitesse de fonctionnement.

Utiliser des boîtiers C-MOS, du type 4050 ou 4049, par exemple, ne convient que dans les applications où l'horloge ne dépasse guère 500 à 600 kHz, malgré un fonctionnement possible à 5 MHz. La raison est liée aux retards dus à la traversée du boîtier, qui avoisinent, dans le meilleur des cas 150 ns en C-MOS, ce qui est prohibitif pour nos mémoires dynamiques, capables d'échantillonner la partie haute des adresses en 30 à 50 ns.

Une règle simple est d'utiliser des boîtiers pouvant fonctionner à 10 fois la fréquence d'horloge du système.

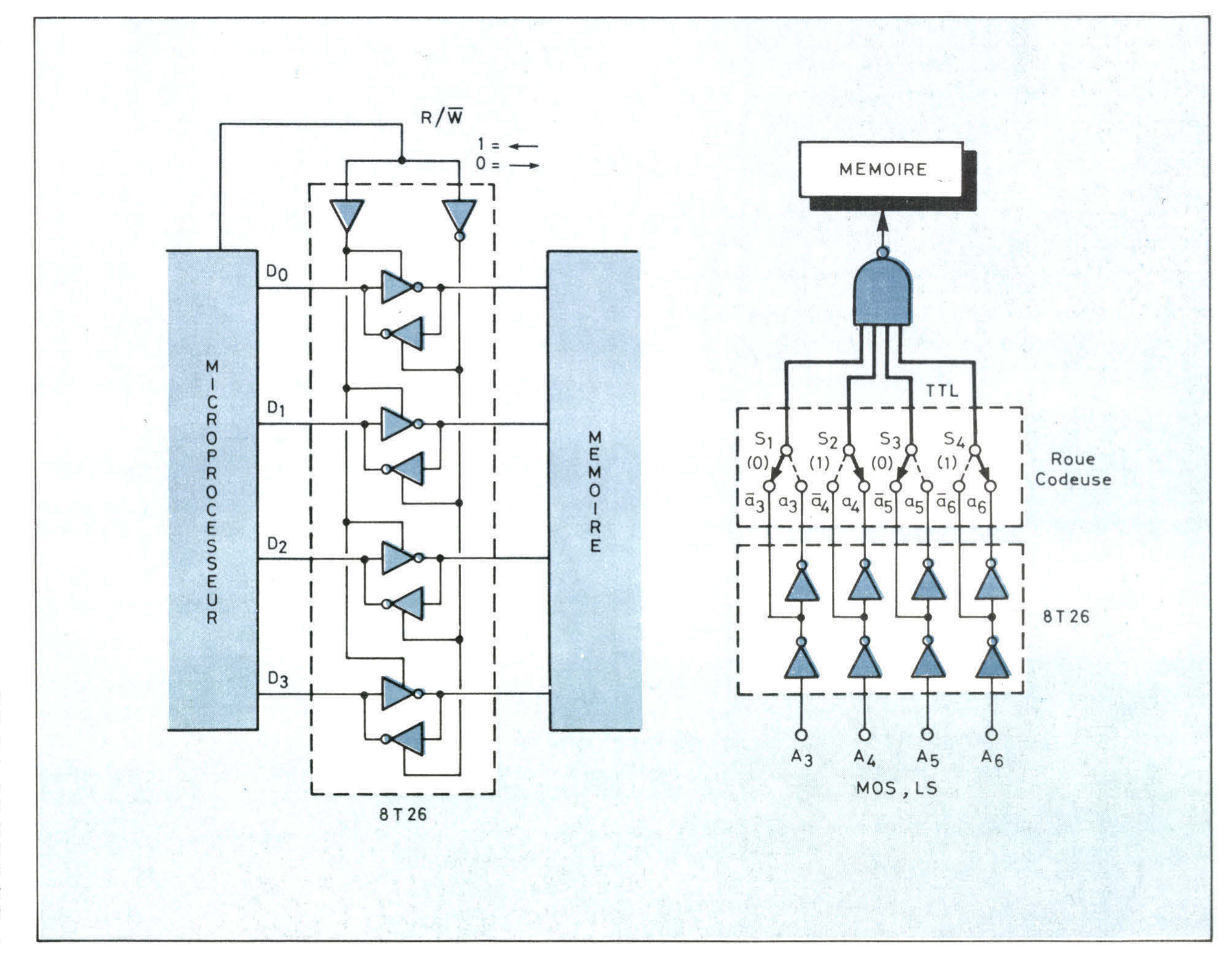


Fig. 3 a. – Exemple de tamponnage d'un bus de données de quatre fils entre une unité centrale et une mémoire. La sélection d'un bloc mémoire sur une adresse déterminée peut être réalisée de façon élégante par des roues codeuses placées à la sortie du tampon 8T26.

Le caractère dynamique des mémoires RAM rend nécessaire leur rafraîchissement à des intervalles de temps réguliers.

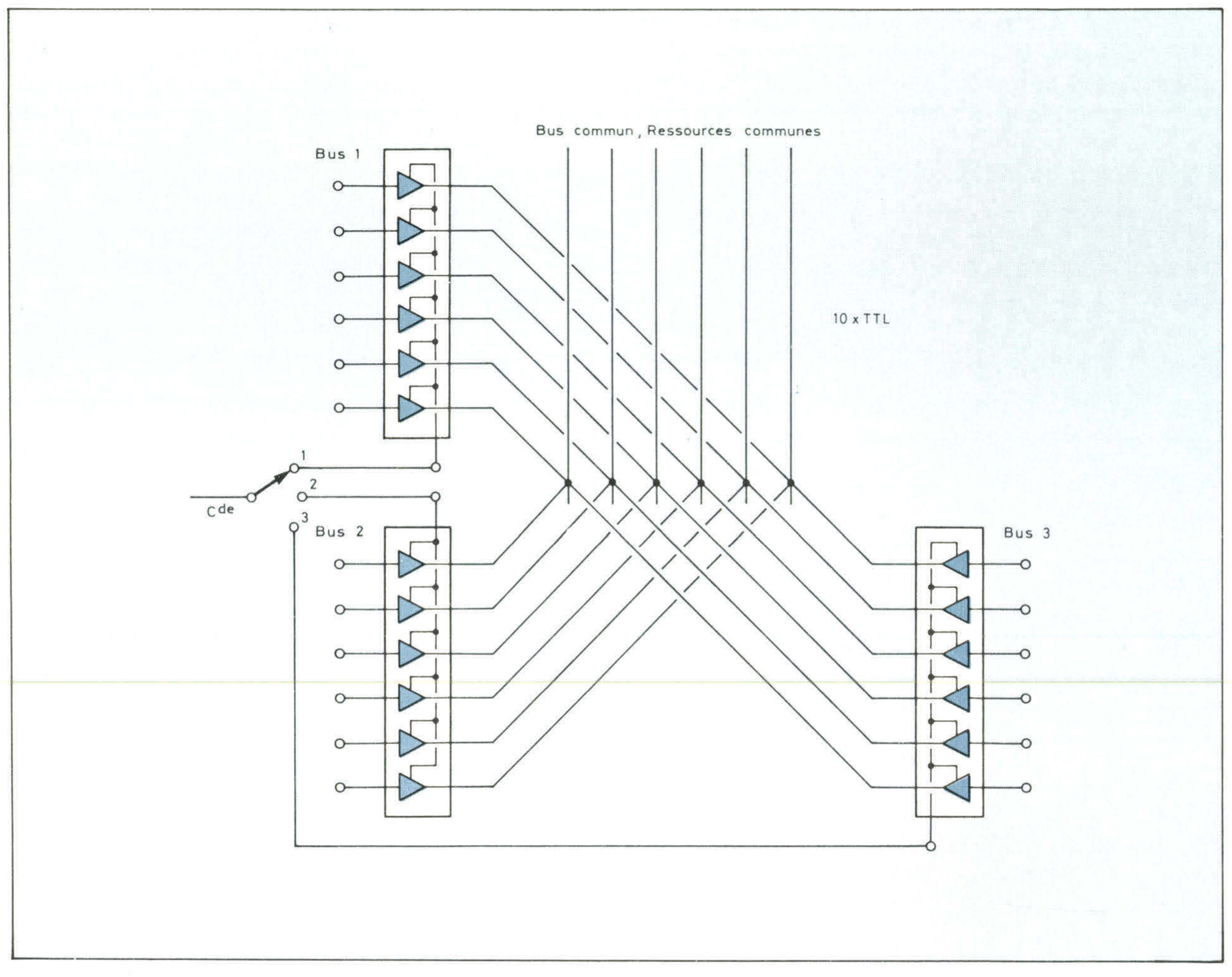


Fig. 3 b. – Le sélecteur de commande place successivement sur les bus 1, 2 et 3 la partie haute et basse de l'adresse et les adresses à rafraîchir.

Une autre règle est de préférer les tampons inverseurs aux non-inverseurs, pour aller vite, car il est connu que pour obtenir une inversion il suffit d'un transistor alors qu'un gain sans inversion s'obtient par la mise en cascade de deux transistors ou plus, en doublant le temps de transfert. C'est d'ailleurs la raison de l'utilisation des boîtiers 8T 26 ou équivalent, malgré l'inconvénient immédiat de devoir employer des bus à signaux inversés.

Un autre problème, d'utilisation pratique lors de la mise au point, est la tenue aux courts-circuits. Il faut pouvoir s'« adapter » à des courts-circuits momentanés, liés à l'ouverture concomitante de deux tampons de ligne sur le même fil. Cela arrive plus souvent qu'on ne le pense et, c'est principalement dû aux capacités parasites et aux petits retards que peuvent prendre les impulsions de commande de tri-state en fonctionnement dynamique. Ainsi, les 8T 95, 97 peuvent supporter des courts-circuits de sortie pendant presque une seconde et débitent en court-circuit 40 à 120 mA par sortie tout en ayant un courant d'entrée de quelques centaines de μA.

Les temps de propagation de tous nos tampons (8T 26, 8T 97, 8T 95) sont de l'ordre de 15 ns.

B.1 La mémoire RAM dynamique

Nous l'avons dit, ce qui distingue notre ordinateur d'un système simple à microprocesseur est la mémoire. Parmi les circuits de **Micro-Systèmes 1** ceux qui assurent le fonctionnement correct des 32 K-octets de mémoire dynamique, avec des prolongements vers les 48 K, sont de loin les plus complexes.

Telle qu'elle est présentée sur le schéma général de la **figure 2,** la partie mémoire présente de nombreuses variables. Il a bien fallu en fixer quelques-unes, concernant surtout les cheminements des parties haute et basse du bus adresses et le rafraîchissement. Le reste a suivi, même en tranchant parfois entre plusieurs variantes équivalentes.

Les boîtiers-mémoire utilisés sont du type SV 6726, mémoires dynamiques de 16 k x 1 bit. Le caractère dynamique de cette mémoire rend nécessaire son rafraîchissement à des intervalles de temps réguliers pour le maintien de l'information stockée **(voir encadré).**

Ce type de mémoire a été choisi pour les raisons suivantes :

- grande densité de bits mémorisés par boîtier et prix de revient du bit très bas ;
- boîtier 16 broches, DIL, facilitant même l'assemblage avec des machines d'insertion automatique,
- compatibilité TTL des entrées et vitesse de fonctionnement compatible avec le reste du système,
- simplicité du rafraîchissement qui ne demande que 128 sélections de ligne toutes les 2 ms.

La **figure 4** présente l'organisation interne d'un boîtier-mémoire de 16 k x 1 bit. On remarque deux entrées de sélection du boîtier, $\overline{RAS}$ (« Row Address Select »), Sélection d'une adresse-ligne et $\overline{CAS}$ (« Column Address Select ») sélection d'adresse-colonne, qui actionnent chacune une horloge interne contrôlant la manipulation des bits de données et les cheminements d'adresse dans le boîtier.

La mémoire proprement dite se présente en deux blocs de 8 K bits de capacité, chaque bloc étant constitué de cellules-mémoire actives et fictives (leurs paires pour des problèmes d'élimination des bruits parasites lors des manipulations à grande vitesse). Sans entrer dans les détails de ces constituants, la matrice mémoire globale, de 16 K bits peut être regardée comme un tableau de 128 lignes sélectionnées par A_7-A_{13} et 128 colonnes, provenant de A_0-A_6, partagées en deux groupes de 64 par le fil d'adresse A_0.

Les adresses sont multiplexées comme dans la plupart des RAM-S de cette capacité : 16 K bits s'adressent par 14 fils. Si l'on tient compte des alimentations et d'un fil d'entrée/sortie du bit de donnée, l'on dépasse les 16 broches des boîtiers DIL bon marché. En réalité nous devons pouvoir disposer de deux broches, d'entrée et sortie du bit de donnée, séparées, ainsi que des broches de sélection du boîtier.

Le multiplexage est régi par $\overline{RAS}$ et $\overline{CAS}$ de la façon suivante :

Le front descendant de RAS permet la réception et l'acheminement vers un registre tampon de la partie haute du bus-adresses A_7-A_{13}, qui adresse les 128 rangées horizontales ou lignes de la matrice-mémoire.

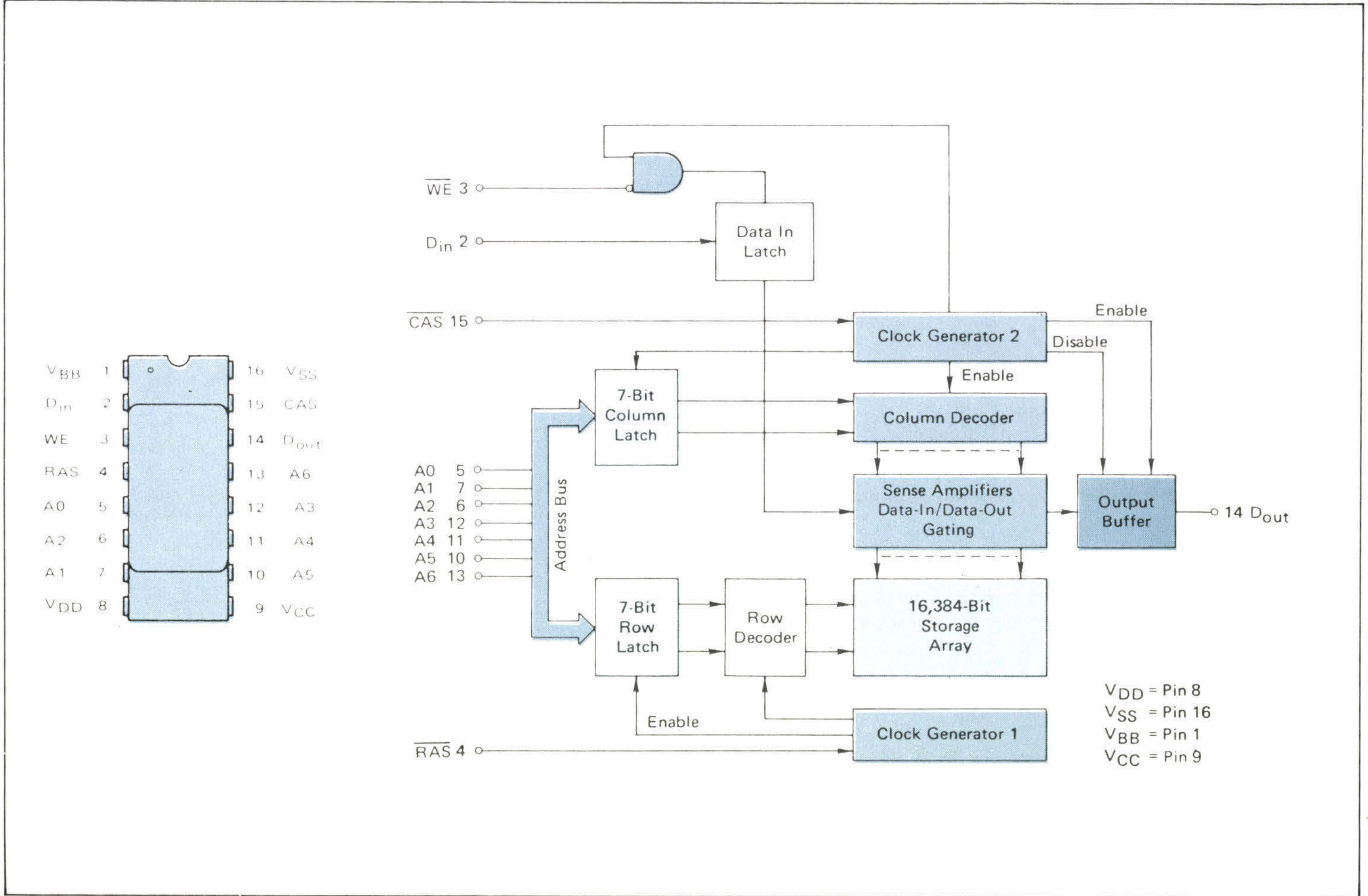


Fig. 4. – Organisation interne et brochage d'un boîtier mémoire de 16 k x 1 bit. Vous pouvez utiliser des SV6726

La descente de CAS valide la sortie des données en même temps qu'il achemine la partie basse des adresses, A_0–A_6 vers les décodages-colonne.

Plus simplement, sur le front descendant de RAS, les sept fils d'adressage de boîtier seront échantillonnés et considérés comme étant la partie haute de l'adresse ; sur le front descendant de CAS, on est en présence de la partie basse.

Le Rafraîchissement de la mémoire s'obtient par l'envoi d'une impulsion $\overline{RAS}$ à toutes les adresses de ligne, une à une, toutes les 2 ms au maximum. Nous sommes très loin de cette limite puisque notre rafraîchissement s'effectue toutes les 12 à 13 μs. 128 impulsions et le balayage dans l'ordre de toutes les adresses de ligne garantissent le maintien de l'information. Pour obtenir ce fonctionnement, une horloge de période 12 μs, dûment validée par l'unité centrale, incrémentera un compteur binaire de 7 étages. A sa sortie on obtient le balayage dans l'ordre des 128 lignes que l'on présentera à l'entrée d'adressage de la mémoire accompagné de $\overline{RAS} = 0$ (vrai) et $\overline{CAS} = 1$ (pas de sélection-colonnes pour le rafraîchissement).

Le temps de l'impulsion de rafraîchissement est sensiblement égal à $\overline{RAS}$ et doit être supérieur à 135 ns. Autrement dit, après avoir présenté pendant environ 500 ns, par exemple, une configuration-adresse allant de 0 à 127 et $\overline{RAS} = 0$; $\overline{CAS} = 1$, la mémoire est disponible à l'utilisation normale jusqu'à l'arrivée de la prochaine impulsion de rafraîchissement (15 μs si on n'a pas de raison de se presser à rafraîchir). On remarque donc un rapport d'indisponibilité, 500 ns/15 μs très faible, 1/30 du temps pouvant être consacré au rafraîchissement dans les cas extrêmes.

N'oublions pas que $\overline{RAS}$ sert en même temps à l'adressage normal de la partie haute des adresses.

L'enchaînement dans le temps de l'adressage haut et bas et du rafraîchissement est résumé sur la **figure 5.**

Il y a d'abord l'horloge de rafraîchissement. Si d'autres demandes (DMA) n'affectent pas l'unité centrale, cette dernière lance une impulsion de rafraîchissement T_A toutes les 13 μs et fait avancer un compteur binaire d'un pas, de sorte qu'au bout de 128 impulsions la combinaison binaire de sortie du compteur repassera par sa valeur initiale. Peu importe quelle est cette valeur, il suffit de parcourir toutes les adresses dans l'ordre à partir de n'importe laquelle (l'entrée R.A.Z. du compteur CD 4024 (U_{19}) est d'ailleurs en permanence à la masse, donc inactivée).

Remarquons deux actions concurrentes : la génération de l'impulsion T_A ($\overline{RAS}$, $\overline{CAS}$) et le positionnement du commutateur électronique sur le compteur CPTR qui doit être à ce moment précis stabilisé, le comptage de l'impulsion devant commencer à la fin de T_A, par exemple.

De la même manière, en dehors du temps de rafraîchissement T_A on doit pouvoir générer le front descendant de RAS et le temps T_B, pendant lequel les adresses hautes seront présentées par le commutateur à la mémoire, suivies de la chute de CAS et du temps T_C, de commutation sur la partie basse des adresses.

Reste donc à construire le commutateur et trouver un séquenceur capable d'assurer le déroulement des opérations dans le temps.

Le commutateur électronique (3 bus de 7 fils vers le bus-adresses d'entrée du boîtier mémoire) est détaillé sur la **figure 6.** La commande des boîtiers-tampon est obtenue par le séquenceur. Remarquez qu'à raison de 6 amplificateurs par boîtier, nos 7 fils nécessitent deux boîtiers-tampon, les 5/6 du deuxième boîtier restant inutilisés.

Les possibilités de commande aux entrées L, M, N du commutateur sont les suivantes :

L'entrée « N » à 1 bloque l'accès à la mémoire des deux moitiés haute et basse du bus adresses A_0-A_{13}. Dans cette position elle permet le rafraîchissement. De plus, son front descendant fait avancer le compteur de lignes à rafraîchir et cette opération a lieu après N-haut, donc après le rafraîchissement, de

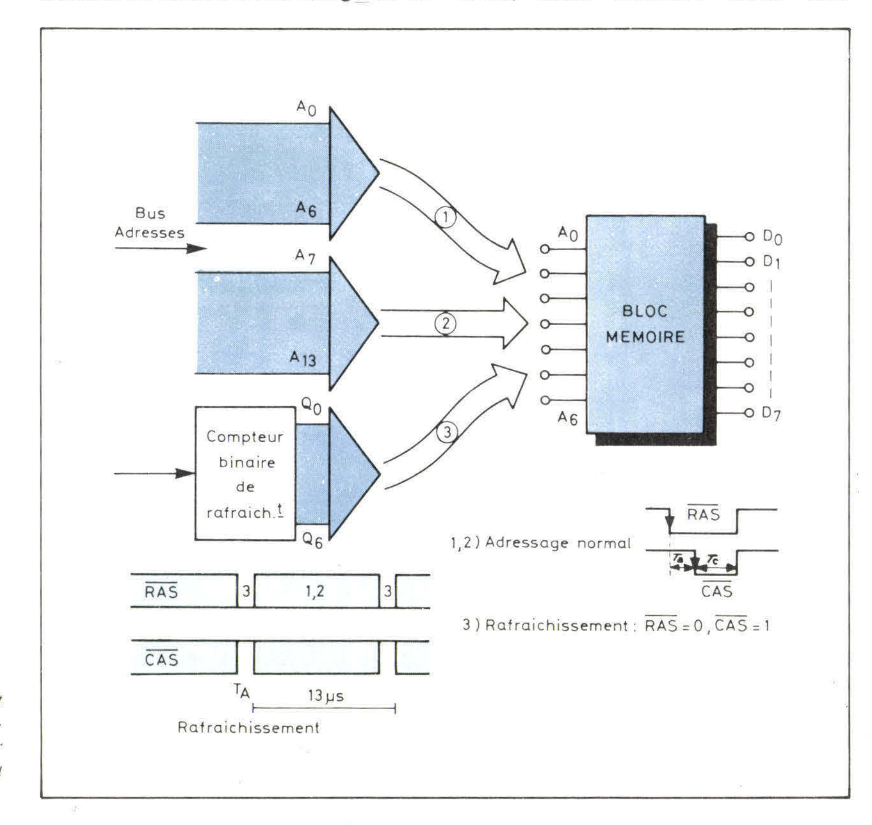


Fig. 5. – Enchaînement dans le temps des informations présentées sur les entrées A_0-A_6 du bloc mémoire.

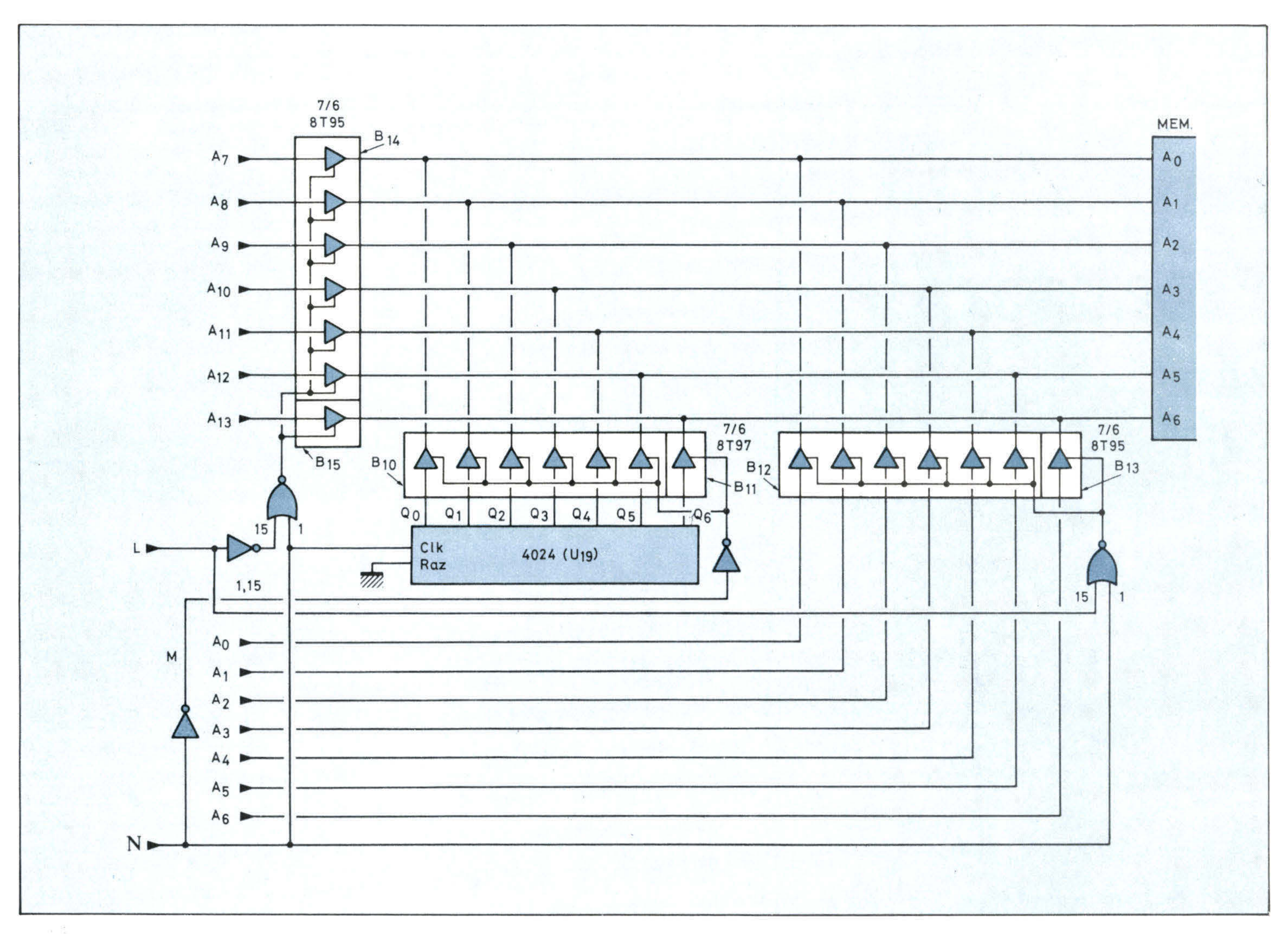


Fig. 6. - Commutation électronique permettant au boîtier mémoire de recevoir successivement la partie haute et basse de l'adresse et l'adresse à rafraîchir.

sorte que le compteur puisse se stabiliser avant une nouvelle phase de rafraîchissement.

L'entrée « L » agit différemment sur la partie haute et sur la partie basse du bus-adresses. A niveau haut et si N est bas, elle valide l'accès de A_7-A_{13} à la mémoire. A niveau bas et N bas, elle valide l'accès de A_0–A_6.

L'entrée M, qui n'est autre que $\overline{N}$, permet d'envoyer le contenu du compteur vers la mémoire.

Remarquons pour finir, que le commutateur est commandé par des niveaux logiques. Sur les diagrammes de la **figure 7a,** « L » peut, naturellement, varier plus d'une fois pendant la marche normale.

Le séquenceur doit assurer, en dernière instance, l'apparition d'un front descendant RAS pendant que les niveaux A_7–A_{13} sont stables en entrée de la RAM et, également, un front descendant CAS après avoir fixé A_0–A_6 cette fois-ci, en entrée.

Pendant le rafraîchissement on doit avoir $\overline{CAS} = 1$ et $\overline{RAS} = 0$.

Les diagrammes de la **figure 7b** résument le fonctionnement des circuits du séquenceur (U_{13}, U_{16}, U_{17}, U_{18}), en marche normale, c'est-à-dire en l'absence de rafraîchissement.

La commutation partie basse/partie haute du bus adresses, L, s'obtient à partir d'une porte NAND (U_{13}-3-4-6), qui reçoit d'une part le signal d'horloge MEMCLK (Ø 2), amplifié, et d'autre part, une inhibition de la part d'un monostable fournissant les impulsions T_1 et $\overline{T_1}$, d'environ 400 ns. Comme Ø 2 $\simeq$ 600 ns, l'impulsion T_1 n'a pas le temps de jouer son rôle d'inhibition, mais la porte est utile car Ø 2 ou MEMCLK sont sujets à des variations dues au fonctionnement en DMA ou avec des horloges externes du système.

Ainsi, le premier front descendant MEMCLK qui trouve le monostable U_{17} au repos, à $\overline{Q}$ = haut, active l'impulsion T_1, $\overline{T_1}$ et bloque pendant la temporisation toute autre entrée vers le point L. Un deuxième monostable, (1/2 U_{18}), fournissant l'impulsion $\overline{T_3}$, est sensible au même front, alors qu'un troisième, produisant $\overline{T_2}$ est sensible au front montant de MEMCLK. Sur les diagrammes de la **figure 7b** on a dessiné L, c'est-

Fig. 7 a. – Etat des signaux L, M et N lors du rafraîchissement et du fonctionnement normal de la mémoire.

Fig. 7 b. – Timing des signaux de sélection de l'adressage.

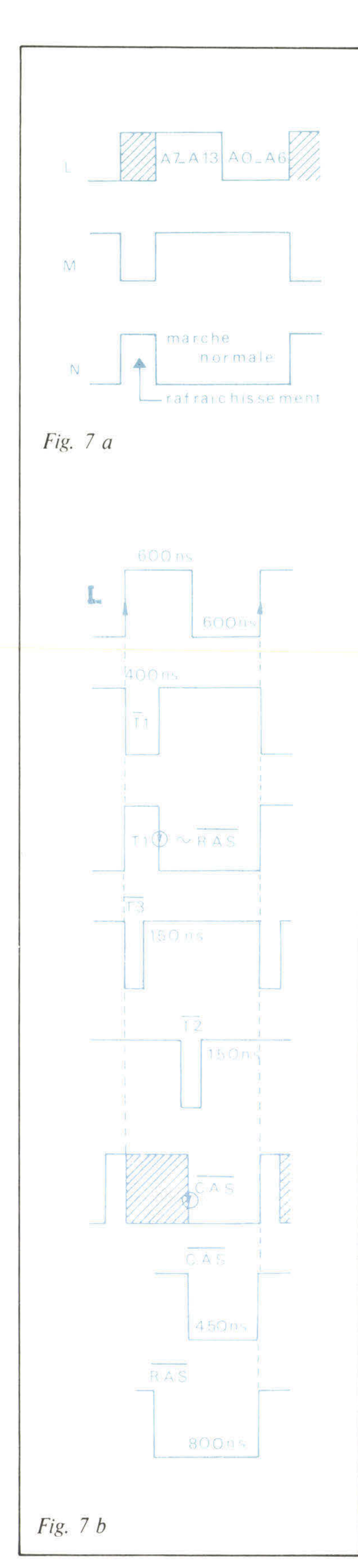


Fig. 7 a

Fig. 7 b

à-dire la sortie 6 de la porte U_{13} et l'entrée de commande des trois monostables.

Le front descendant de T_1 ne produit pas directement $\overline{RAS}$, car, faute d'autres sélections, nous devons positionner les blocs-mémoire dans l'espace adressable par les mêmes $\overline{RAS}$, $\overline{CAS}$. Alors, compte tenu de la fiche technique, les 32 K octets-mémoire, placés aux adresses

0 0 0 0 à 7 F F F

demandent une sélection globale par $A_{15} = 0$. Pour départager le premier bloc de 16 K octets (0 0 0 0 à 3 F F F) du second (4 0 0 0 à 7 F F F), nous devons les sélectionner par :

$A_{14} = 0$ 0 0 0 0 à 3 F F F
$A_{14} = 1$ 4 0 0 0 à 7 F F F

L'introduction de ces sélections d'adresse dans la fabrication des impulsions $\overline{CAS}$ et $\overline{RAS}$ s'obtient de la manière suivante :

• A_{14}, A_{15}, $R/\overline{W}$, VMA * arrivent par les bus à l'entrée des tampons B_9. Comme pour MEMCLK (Horloge mémoire) et REFGNT (Accusé de Réception de la demande de rafraîchissement), ces amplificateurs sont activés en permanence (1,15 à la masse). Un latch à 4 bits, 7475 validé par le front descendant de T_1 fournit A_{14}, $\overline{A_{14}}$, A_{15}, VMA et $R/\overline{W}$ pendant la période d'horloge, jusqu'à l'arrivée d'un autre front descendant T_1.

La commande de lecture/écriture $R/\overline{W}$ sera envoyée pendant $\overline{CAS}$ à l'entrée $\overline{W}$ des mémoires.

Le signal VMA joue un rôle important et sert au même titre que A_{14}, A_{15} à l'adressage : les bus du microprocesseur ne sont pas actifs durant toute la période d'horloge. A l'état haute impédance les tensions des niveaux logiques sur le bus « flottent » au gré des capacités parasites et peuvent parfaitement actionner dans un sens ou dans l'autre des dispositifs dont l'entrée MOS garantit une extraordinaire sensibilité. Nombre de montages fonctionnent dans des configurations plus ou moins idéales grâce justement aux capacités parasites et à la persistance des adresses souvent présentes au-delà des durées des bus actifs.

Pour échapper à ces doutes, nous allons utiliser l'indication de validation mémoire (VMA) dans les circuits de sélection. Par un circuit ET U_{11} (7411), nous fabriquons le signal $\overline{A_{15}}$.VMA qui est envoyé d'une part à l'entrée D d'un bistable 7474 de U_{16} (qui produit $\overline{CAS}$ et $\overline{W}$) et d'autre part, aux portes U_{37} servant à la fabrication de $\overline{RAS}$. Les deux blocs de 16 K auront un $\overline{RAS}$ sélectionné par A_{14} et $\overline{A_{14}}$ aux portes U_{37} et l'on obtient $\overline{RAS_2}$ et $\overline{RAS_1}$, respectivement.

Les mêmes signaux $\overline{A_{15}}$.VMA et $R/\overline{W}$ servent au positionnement des tampons du bus de données vers la mémoire.

Le bit de donnée de chaque boîtier RAM-dynamique est dirigé vers la paire d'amplificateurs B_4, B_5.

L'inverseur U_{12} pourrait être remplacé par l'utilisation de la sortie 10 de U_{15} (7475), mais il resterait néanmoins inutilisé dans le boîtier qui en contient six. Par cet inverseur et la porte U_{13} on crée le signal $\overline{A_{15}}$.VMA. $R/\overline{W}$ qui, inversé par U_{12} commande le sens d'amplification de B_4 et B_5.

Le rafraîchissement exige deux signaux : $\overline{REFREQ}$ est la demande de rafraîchissement, active à l'état bas, qui émane du monostable U_{18} fournissant T_4. Cette temporisation, relativement longue aboutit, à niveau bas au signal de demande. Ce signal arrive à l'entrée $\overline{DMA}$ ou $\overline{REF\ REQ}$ de l'horloge microprocesseur, qui bloque les phases d'horloge du microprocesseur, $\varnothing_1$ à l'état haut et $\varnothing_2$ à l'état bas pour permettre le rafraîchissement. Lors de ce blocage se produit une différence entre $\varnothing_2$ et MEMCLK, cette dernière continuant son cycle initial. Au blocage d'horloge s'ajoute la montée à niveau haut d'une sortie DMA ou REF GNT, l'accusé de réception de la demande de rafraîchissement. Ces opérations ont pour mission d'éviter, par le blocage de l'horloge, le déroule-

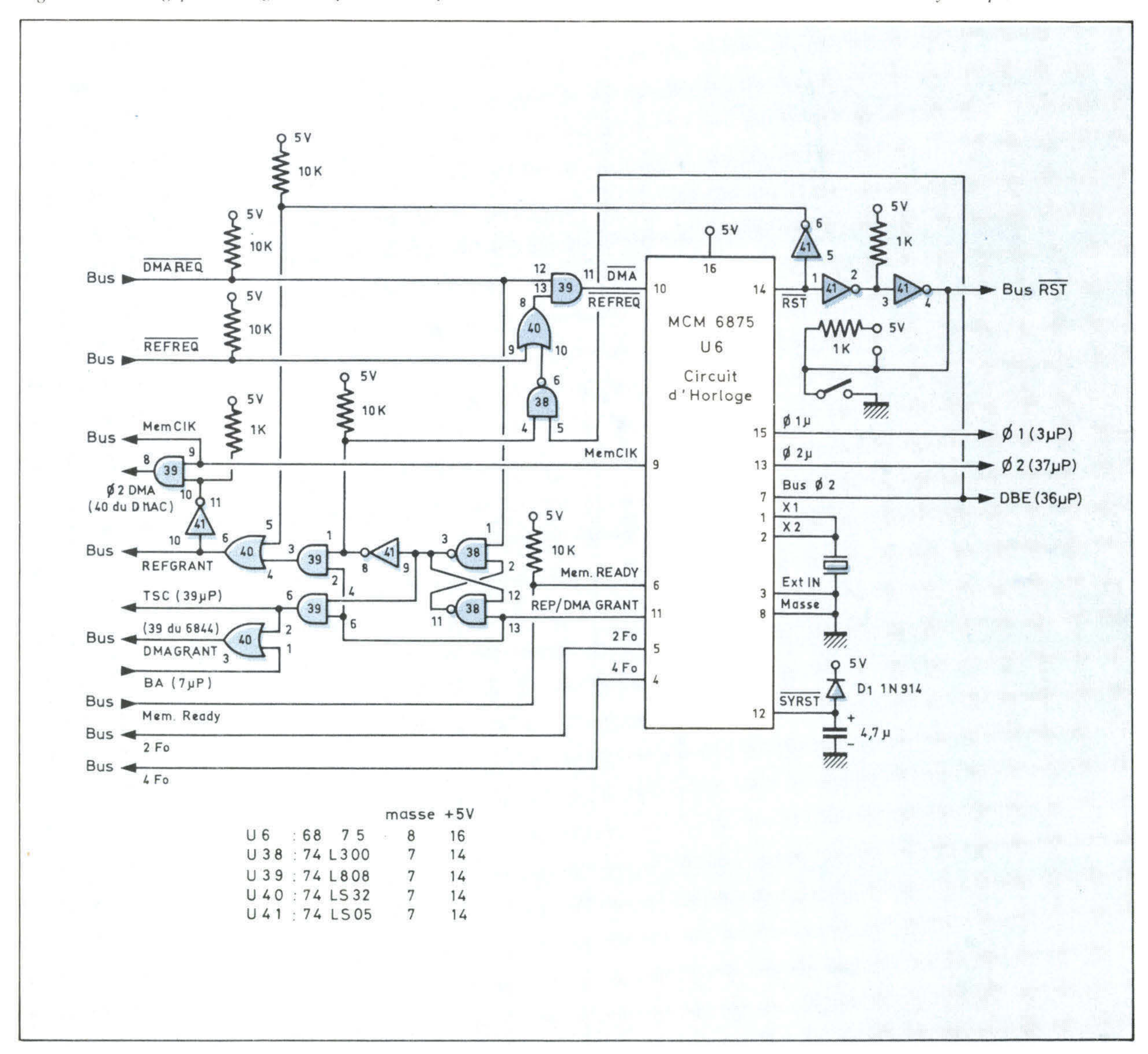
* *VMA : Valid Memory Address : adresse mémoire validée.*

ment des programmes qui pourrait aboutir à une écriture ou lecture de la mémoire en cours de rafraîchissement. A la réception du niveau haut REFGNT, préalablement amplifié par le tampon B_{11} au même titre que MEMCLK, un latch U_{14} (7475) transmettra au bout d'un temps T_1, sur le front descendant, ce niveau haut au compteur de rafraîchissement, U_{19}, positionnant les tampons du compteur, commandés au point M, après inversion et à l'entrée du monostable U_{18} qui fournit la temporisation T_4 entre deux rafraîchissements.

Remarquons que le compteur et le monostable sont sensibles au front descendant de l'horloge. Grâce à l'inverseur le monostable se positionnera dès l'arrivée de REF GNT fournissant la commande du point N. Ceci mettra fin, par l'apparition du niveau haut T_4, à la demande de rafraîchissement. La réponse REF GNT reste à niveau haut, pendant environ un cycle d'horloge et reproduit la période de MEMCLK. Si la demande de rafraîchissement est asynchrone, déterminée par T_4, la réponse est synchrone au MEM CLK et nous le mettons à profit pour la bonne marche des circuits.

Comme toujours, il est plus facile de faire et de rajouter en électronique que d'expliquer avec luxe de détails. Ces circuits peuvent s'améliorer sans cesse. Nous les avons figés dans cet état précis et ils ont été implantés sous cette forme.
Le compteur de rafraîchissement et l'électronique associée peuvent servir à rafraîchir de la RAM supplémentaire, située par exemple sur les connecteurs enfichables, perpendiculairement à la plaque-mère. L'adresse-mémoire de cette RAM pourra être quelconque. Seuls changeront les circuits addi-

Fig. 8. – Circuit logique d'horloge utilisé pour le microprocesseur, l'accès direct à la mémoire (DMA) et les mémoires dynamiques.



Les RAM dynamiques et leur rafraîchissement

Une mémoire de M.mots de N.bits chacun contient M x N cellules élémentaires. Ces cellules, permettant l'inscription et la lecture d'une information de 1 bit, peuvent être de plusieurs sortes : dans les mémoires vives (RAM), dites statiques, la cellule élémentaire est composée d'un bistable, c'est-à-dire d'un circuit électronique, composé principalement de deux transistors (T_1 et T_2), pouvant prendre deux états stables. Ces deux états sont : T_1 en conduction, T_2 bloqué et inversement T_1 bloqué, T_2 en conduction.

En conduction, chaque transistor consomme un courant et les résistances de drain dissipent une puissance faible, mais non-négligeable si l'on pense que la pastille de silicium en contiendra quelques milliers, à raison de deux résistances par cellule. Une RAM utilisant de tels bistables, appelés « MOS Canal-N (ou P) » consommera dans l'ensemble, pour 4096 bits de capacité (autant de bistables), environ 100 mA sous 5 V.

Pour éviter la consommation au repos du bistable, les ingénieurs ont imaginé d'utiliser à la place des résistances de drain, des transistors MOS complémentaires (C.MOS).

On constate que quelque soit l'état du système, il y aura un transistor qui bloquera le passage du courant. On arrive en définitive à obtenir une réduction de la consommation de l'ensemble d'un facteur de 1 000, des consommations globales de 10 ou de 100 mA devenant courantes.

Malheureusement, un transistor MOS n'est pas exempt d'une capacité parasite d'entrée-grille/source non négligeable (qui double si l'on utilise la technologie MOS-Complémentaire caractérisée par l'attaque de deux transistors, donc de deux grilles à la fois).

Mais cette capacité parasite, pourrait être utilisée à profit comme élément mémoire, ce qui a donné naissance à un deuxième type de RAM-S.

Dans les RAM-S dynamiques l'élément mémoire n'est plus un bistable mais la capacité parasite grille-canal d'un transistor MOS, T_1 sur la **figure A.** Cette capacité est de l'ordre de quelques pF. La résistance d'entrée de T_1 étant de quelques GΩ (10^9 Ω), la constante de temps de décharge de C_1 avoisine la dizaine de millisecondes.

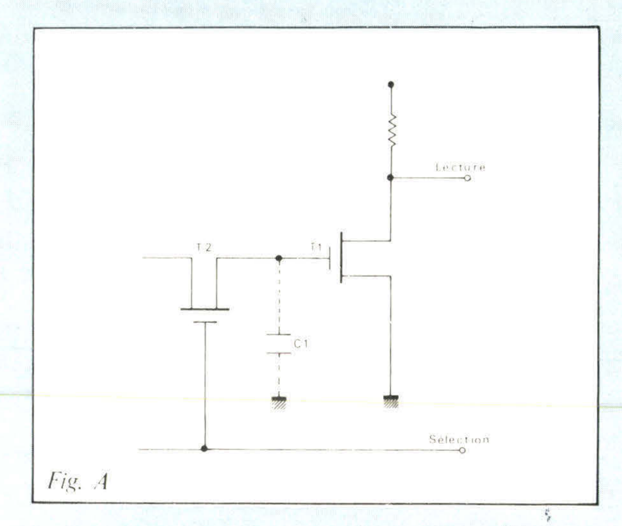


Fig. A

Donc, si par T_2 on charge la capacité C_1 à la tension + V et si par la suite on bloque T_2, la charge du condensateur se maintiendra pendant environ 10 ms. Le fait d'être chargé ou non constitue la mémorisation d'un 1 ou d'un 0.

Pour maintenir l'état de charge de C_1, on utilise la technique du rafraîchissement qui consiste en un envoi d'impulsions de recharge vers les condensateurs qui sont déjà chargés. Si l'on transmet ces impulsions toutes les 2 ms, au plus, par exemple, la tension aux bornes de C_1 restera en permanence dans la zone du niveau. « 1 » logique. Si, par mégarde, on oublie de rafraîchir une seule fois, la mémoire se volatilise, même sans coupure de l'alimentation.

En réalité, le fonctionnement à grande vitesse exige des capacités-mémoire les plus petites possibles, ce qui rend les cellules-mémoire très sensibles aux bruits et parasites. Pour mieux faire, au lieu d'amplifier par un transistor T_1 la tension d'un seul condensateur-mémoire C_1, on est amené dans la pratique à utiliser deux condensateurs pouvant recevoir les parasites en commun, mais dont seulement un reçoit les éléments à mémoriser. Par la suite on utilise une amplification différentielle des tensions de la cellule mémoire proprement dite et de la cellule paire ou fictive et l'on élimine ainsi les bruits, les dérives thermiques, les parasites, etc.

tionnels de positionnement-mémoire équivalents à nos portes U_{11}, U_{37} qui manipulent les fils d'adressage A_{14}, A_{15}.

B.2. L'unité centrale, le contrôleur de DMA et l'horloge

Les ressources du système, mémoires, entrées/sorties, etc, peuvent être gérées par deux « maîtres », dépendant tous les deux du générateur d'horloge de la **figure 8.** Les deux utilisateurs possibles ont en commun la qualité **d'émetteurs d'adresse** alors qu'aucun autre circuit dans le système ne peut le faire. Il s'agit de les soumettre à un certain protocole pour éviter les conflits. Ce protocole peut passer par l'« intelligence » du boîtier unité centrale 6800, quand la passation des pouvoirs se fait par une action sur l'entrée « Halt » ou par l'allongement des cycles d'horloge, prévu dans le fonctionnement du circuit d'horloge 6875, auquel cas on fait du « vol de cycle », l'unité centrale ne s'apercevant de rien.

Ainsi les trois boîtiers, le 6875-générateur d'horloge, le 6800 (unité centrale) et le 6844 (contrôleur de DMA), interdépendants, donnent au système la possibilité de fonctionner avec ou sans unité centrale, en configuration multi-micro-ordinateurs ou monoposte, laissant prévoir la possibilité de connexion à des mémoires de masse comme les disques souples ou durs, etc.

Conclusion

Après cette présentation de la partie mémoire et des possibilités offertes par la présence d'un contrôleur de DMA sur la carte, nous passerons en revue, dans notre prochain numéro les autres ressources du système : interfaces vidéo, clavier, minicassette et nous décrirons l'interpréteur BASIC. D'ores et déjà on peut remarquer que toutes les ressources de la plaque pourront être mises à la disposition d'un autre micro-ordinateur extérieur par l'intermédiaire du contrôleur de DMA.

Bien entendu, nous vous présenterons de façon détaillée le cœur du système : le microprocesseur 6800. ■

A. DORIS

MICRO/EXPO 79
PARIS

15 - 17 MAI

PALAIS DES CONGRES
4E SALON ANNUEL DES MICROORDINATEURS

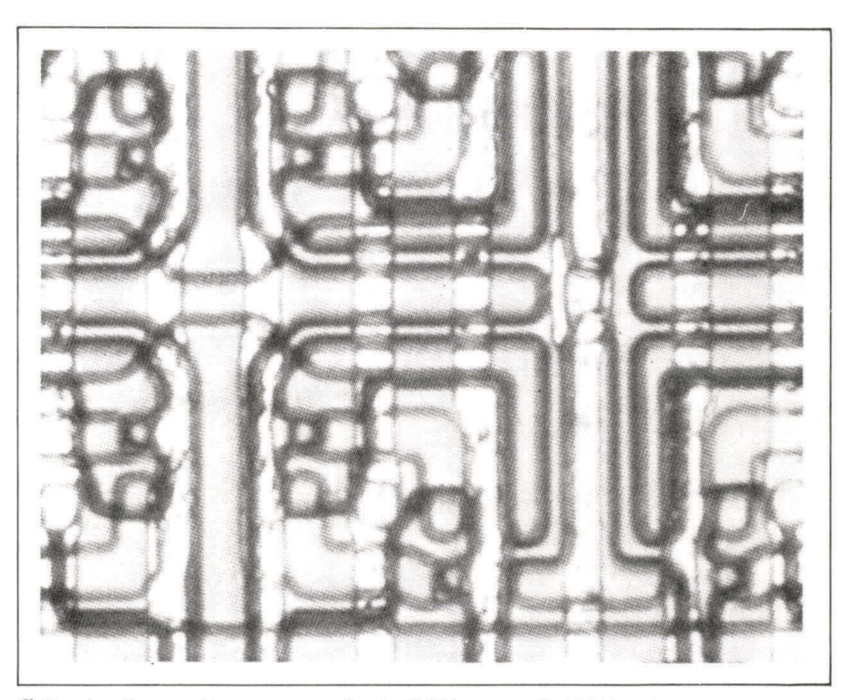


Renseignements : **SYBEX, 313, rue Lecourbe, 75015 Paris - Tél. : 828-25-02 et 531-21-53**

La technologie H.MOS

C'est presque devenu une loi de la nature : les circuits métal-oxyde-semiconducteur doublent de densité ou de performance chaque année. Au cours des dix dernières années, les circuits MOS se sont développés à partir de registres à décalage, de portes, de bascules, fonctionnant à des vitesses de l'ordre de la milliseconde, pour devenir des mémoires complètes, des microprocesseurs, et des systèmes et sous-systèmes spécialisés intègrant des dizaines de milliers de fonctions électroniques dans un seul organe capable de travailler à la nanoseconde.

Le moteur de cet étonnant progrès est l'extraordinaire polyvalence de la technologie MOS. Au départ, simple technique de circuit à alimentation multiple en canal-P capable de gérer des fonctions élémentaires de calcul ou de mémorisation en série, le MOS est passé aux structures à simple et à double couche en canal N faisant appel à une alimentation unique de 5 volts. Il exécute actuellement des instructions d'ordinateur complexes avec des temps de cycles de moins de 100 ns pour une puissance dissipée de plus en plus faible.



Cette microphotographie montre une mémoire RAM statique de 4 096 bits. La longueur du canal de chaque transistor MOS est de 4 µm.

Pour la première fois la technologie MOS est capable de rivaliser en performance avec les circuits bipolaires. Les techniques qui se sont montrées capables d'accomplir cette percée sont variées ; mais elles ont toutes en commun une caractéristique essentielle : elles raccourcissent la longueur du canal ou l'espacement drain-source du transistor MOS initial.

Deux approches sont possibles pour améliorer les performances des MOS. L'une repose sur un processus de double diffusion : un élément à déplétion (appauvrissement) avec un canal relativement long (5 microns) sous la porte MOS, est intégré en série avec un canal à enrichissement de 1 µm, formé par diffusion de bore à travers l'ouverture auto-alignée de la jonction source. Le processus est appelé **D-MOS** quand la structure de double diffusion a une configuration planaire, et **V-MOS *** quand la structure a une configuration verticale, dans laquelle la surface du transistor MOS est déposée sur un sillon en forme de V gravé de façon anisotrope dans le substrat de silicium. Dans l'un ou l'autre cas, la double diffusion exige une technologie de traitement nouvelle, et la structure des circuits diffère notablement de celles que l'on trouve dans les techniques de la porte au silicium, classiques.

** Les V.MOS : Micro-Systèmes n° 1.*

L'autre approche repose sur une réduction d'échelle appliquée directement à la taille et aux paramètres des circuits MOS — autrement dit, sur la réduction des structures à portes au silicium en canal N classiques. Cela n'a rien de nouveau ; dès l'origine, ceux qui ont conçu le MOS savaient qu'en réduisant la taille de leurs circuits, ils pouvaient obtenir des vitesses plus élevées, des densités plus fortes, et une puissance dissipée plus faible.

Pour mettre au point cette nouvelle technologie appelée H-MOS, INTEL a choisi la méthode directe de réduction d'échelle, pour deux raisons. En premier lieu, elle découle directement du traitement classique des circuits à porte au silicium et n'exige par conséquent ni nouvelles structures d'éléments, ni schémas de circuits complexes (l'une ou l'autre exigence aurait rendu la production et les coûts de fabrication trop imprévisibles pour garantir leur utilité sur une large gamme de produits semiconducteurs). En second lieu, elle nécessite des dessins de circuit de plus en plus petits qui peuvent être réalisés par les méthodes photolithographiques plus raffinées et par les techniques de fabrication des « wafers » par faisceau d'électrons.

En outre, les structures à double diffusion ont un avenir limité. Elles pouvaient convenir il y a deux ou trois ans, quand l'industrie était incapable de fabriquer des canaux de moins de 5 ou 6 µm de long. Mais maintenant que des longueurs de canaux de 4 µm (et bientôt de 3 et 2 µm) sont accessibles, la nécessité de nouvelles structures comme le D-MOS ou le V-MOS ne sont peut-être plus nécessaires.

Comment changer l'échelle d'un circuit MOS

La **figure 1** montre la coupe d'une porte au silicium en canal N, où L désigne la longueur du canal, T_{ox} l'épaisseur d'oxyde sous la porte (oxyde de porte), X_J la profondeur de la jonction, L_D la diffusion latérale, et C_B le niveau de dopage du substrat. La théorie au premier ordre indique que les

En réduisant les dimensions d'un élément MOS, le concepteur à un moyen d'action considérable sur la densité et sur les performances du circuit.

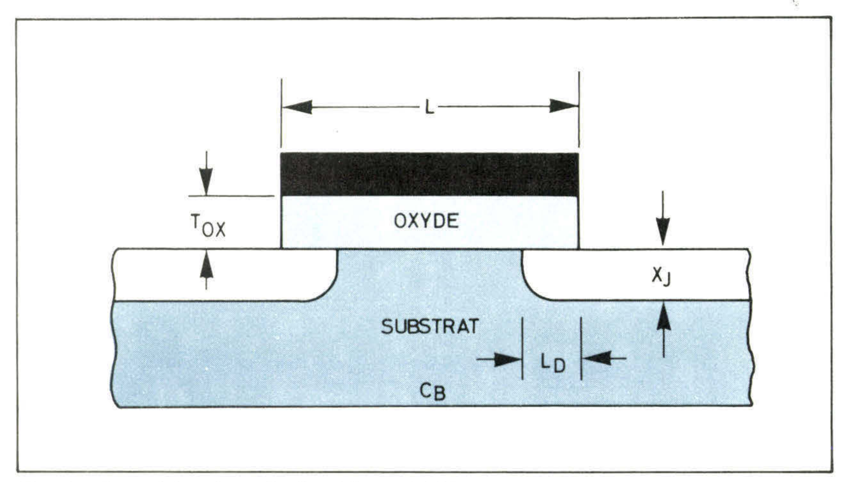


Fig. 1. – Pour réduire la taille d'un élément MOS, tous les paramètres physiques doivent être réduits proportionnellement. Si la longueur L du canal est multipliée par 1/S où S est le facteur de réduction, alors, l'épaisseur d'oxyde T_{OX}, la sous-diffusion latérale L_D, et la profondeur de jonction X_J doivent également être réduits par un facteur de 1/S. Dans le même temps, la constante de dopage du substrat C_B doit être multipliée par S.

TABLEAU 1

CHANGEMENT D'ECHELLE D'UN ELEMENT M.O.S.

Paramètre élément/circuit	Facteur d'échelle
Dimensions de l'élément T_{ox}, L, L_D, W, X_J	1/S
Dopage du substrat C_B	S
Tension d'alimentation V	1/S
Courant d'alimentation I	1/S
Résistance de capacité parasite WL/T_{ox}	1/S
Délai de porte VC/I (τ)	1/S
Puissance dissipée VI	$1/S^2$
Produit puissance-délai	$1/S^3$

TABLEAU 2

EVOLUTION DE L'ECHELLE DES ELEMENTS M.O.S.

Paramètres élément/circuit	N-MOS à enrichissement 1972	N-MOS à déplétion 1976	H-MOS 1977	MOS 1980
Longueur de canal L (μm)	6	6	3,5	2
Diffusion latérale L_D (μm)	1,4	1,4	0,5	0,4
Profondeur de jonction X_J (μm)	2	2	0,8	0,8
Epaisseur d'oxyde de porte T_{ox} (Å)	1.200	1.200	700	400
Tension d'alimentation V_{CC} (V)	4-15	4-8	3-7	2-4
Délai de porte minimum (ns)	12-15	4	1	0,5
Puissance de porte P_D (mW)	1,5	1	1	0,4
Produit vitesse-puissance (pJ)	18	4	1	0,2

caractéristiques d'un circuit MOS peuvent être maintenues et le fonctionnement assuré, si les paramètres du circuit changent d'échelle comme le montre le **tableau 1.** Si S est le facteur d'échelle, et que la longueur du canal L est réduite d'un facteur 1/S, toutes les autres dimensions du circuit — l'épaisseur de l'oxyde de porte et la sous-diffusion latérale, la largeur du circuit et la profondeur de la jonction — doivent également être réduites par 1/S. De plus, pour maintenir une tension de seuil et une tension drain-source de claquage satisfaisante, la concentration de dopage du substrat doit être multipliée par S, tandis que la tension et l'intensité d'alimentation doivent décroître d'un facteur 1/S.

L'effet sur la performance

Si cela est fait convenablement, l'accroissement en performance du circuit est spectaculaire, comme le montre également le **tableau 1.** La capacité parasite, le temps de propagation à travers la porte (délai de porte), la puissance dissipée et le produit délai-puissance dissipée s'améliorent de façon notable. Comme la capacité diminue approximativement avec l'épaisseur de la jonction, elle est réduite, aussi d'un facteur 1/S ; cela signifie que, puisque le délai de porte est à peu près proportionnel à la capacité, il est lui-même réduit par un facteur 1/S. De plus, comme la puissance dissipée par le circuit est proportionnelle à la tension et à l'intensité d'alimentation, elle est réduite par un facteur encore plus élevé en $1/S^2$. En fin de compte, le produit délai-puissance dissipée ou facteur de qualité du circuit MOS (produit du délai de propagation à travers la porte par la puissance dissipée) est réduit par un facteur très important, du type $1/S^3$. Ainsi, la réduction des dimensions d'un circuit MOS améliore ses performances par le cube du facteur de réduction.

En bref, en réduisant les dimensions d'un élément MOS, le concepteur a un moyen d'action considérable sur sa densité et sur

les performances du système (**tableau 2**). On remarquera la nette réduction des paramètres du circuit intervenue entre 1976 et 1977, lorsque Intel est passé au H-MOS à partir de la porte au silicium en canal N classique. En réduisant la longueur du canal de 6 à 3,5 μm et en faisant décroître les autres paramètres en conséquence, il a été possible de diviser le produit vitesse-puissance par quatre. Cette amélioration aurait encore été plus importante si la tension d'alimentation avait été réduite comme l'exigeait la théorie ; au lieu d'être maintenue au niveau de 5 V mieux accepté par les utilisateurs. Cependant, en 1980, lorsque la longueur du canal descendra à 2 μm et la tension d'alimentation à 3 V, les performances s'amélioreront d'une façon encore plus spectaculaire pour devenir au total 20 fois meilleures que celles des circuits MOS de 1976.

Un exemple de ce que la réduction d'échelle représente pour l'utilisateur de circuits intégrés est donné aux **figures 2** et **3** qui retracent graphiquement les progrès réalisés au cours des années dans le domaine des mémoires vives (RAM) statiques. En 1972, les RAM statiques standard en MOS étaient les 2102 à 500 ns, construites avec une longueur de canal de 6 μm et une épaisseur d'oxyde de porte à 1 200 angstroms.

Le produit vitesse-puissance résultant était de 18 picojoules. Elles occupaient une puce de silicium d'environ 3,5 mm de côté et la cellule élémentaire avait une surface de près de 5 000 μm^2.

En 1974 la 2102 a été conçue autour d'une technologie en canal N appauvrie réduisant la surface de base de 15 % et diminuant son temps d'accès jusqu'à 200 ns. A ce processus, l'isolation par oxyde et la polarisation préalable du substrat furent ajoutées en 1976, pour créer la RAM statique 2115 de 1 024 bits d'un temps d'accès de 70 ns. Aujourd'hui, l'incidence de la réduction d'échelle est encore plus évidente avec la technologie H-MOS adaptée aux RAM 2115 (appelées maintenant 2115 A)

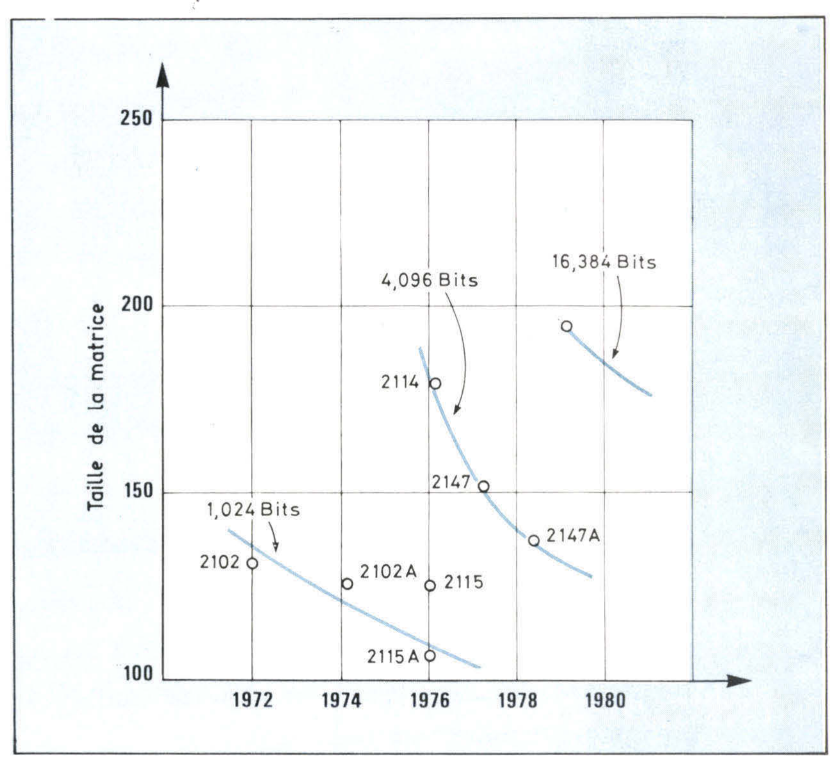
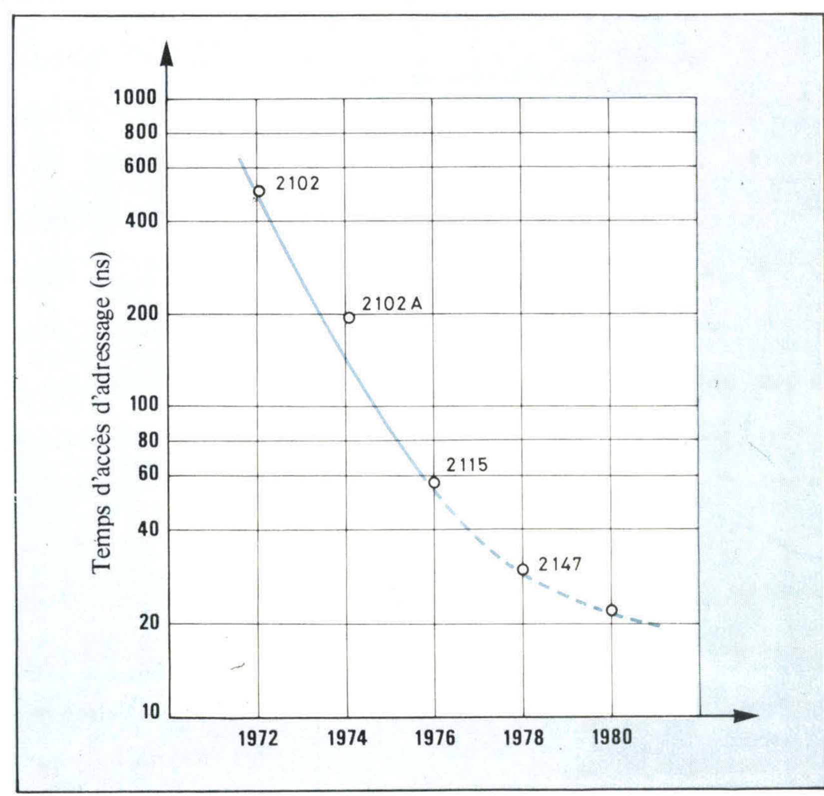


Fig. 2. – Grâce au développement rapide de la technologie MOS, les mémoires vives statiques s'améliorent continuellement en densité. Le H.MOS actuel accroît la densité d'intégration d'un facteur 4.

Fig. 3. – La réduction d'échelle telle qu'elle est réalisée dans le H.MOS rend également les mémoires vives plus rapides. En 1972, une mémoire vive statique type 2102 de 1024 bits avait un temps d'accès de 600 ns ; les mémoires 2147 à 4096 bits actuels ont un temps d'accès de 45 ns.



puisqu'elles sont réalisées sur une puce de 2,5 mm de côté pour un temps d'accès de 25 ns.

De plus, appliqué à une mémoire statique de 4 096 bits, le H-MOS donne une puce légèrement plus grande que le 2102 original, en ramenant cependant le temps d'accès au-dessous de 50 ns. Finalement, à mesure que le processus MOS évolue et que la réduction d'échelle se poursuit, une mémoire vive statique de 16 384 bits tiendra dans une puce qui n'aura pas plus de 5 mm de côté et qui offrira aux concepteurs de systèmes des temps d'accès de l'ordre de 50 ns.

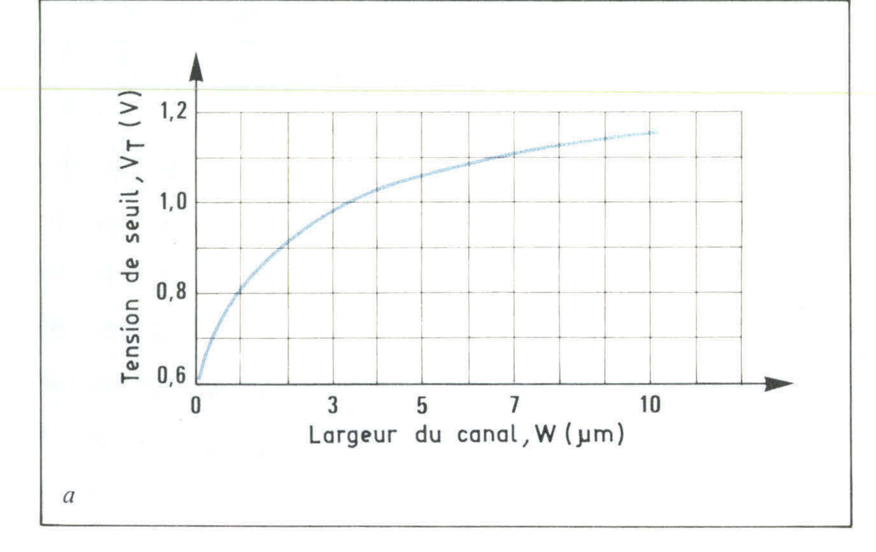
La grande vitesse et la haute densité des H-MOS sont obtenues grâce à cinq améliorations majeures dans la technologie MOS, parmi lesquelles quatre sont directement liées à la réduction d'échelle. Un substrat à haute résistivité est employé pendant que la théorie est appliquée à l'épaisseur d'oxyde sous la porte, à la profondeur de jonction, à la longueur de porte et aux implantations ioniques qui modifient la tension de seuil.

Le substrat de haute résistivité réalisé avec un matériau de type P de 50 Ω-cm est utilisé pour diminuer la capacité de la jonction, réduire les effets de structure du substrat qui dégradent les performances et augmenter la mobilité effective des porteurs. Ces trois facteurs conduisent à des circuits plus rapides et diminuent la consommation en puissance.

En réduisant l'épaisseur d'oxyde de porte du MOS (700 angströms), on accroît le gain du circuit et l'on diminue les effets de substrat et de canal court, augmentant ainsi la performance et la fiabilité. La profondeur de jonction est réduite approximativement à 0,75 μm par l'emploi d'arsenic à diffusion lente comme dopant de source et de drain. La faible profondeur des jonctions augmente à la fois la vitesse en abaissant la capacité des jonctions, la capacité Miller entre la porte et le drain, et la densité en permettant un espacement plus faible entre les zones de diffusion.

Enfin, la stabilité de la tension de seuil du H-MOS est assurée aussi bien en mode enrichissement qu'en mode déplétion, par une zone formée par implantation ionique, utilisée conjointement avec un substrat à forte résistivité. Ce procédé d'implantation permet de maîtriser la tension de seuil avec une grande précision et offre la possibilité de l'optimiser indépendamment du dopage du substrat.

Heureusement, il s'est avéré possible de mettre au point la technologie H-MOS en un temps relativement court (environ un an), sans affecter la capacité de fabrication des éléments à un coût raisonnable. Le H-MOS est également assez souple pour être employé dans une large variété de formes de circuits tout en conservant ses qualités : grande vitesse, petite taille et faible puissance. Contrairement à la logique V-MOS et à la logique à injection intégrée qui exigent de nouvelles techniques de circuit pour être utilisables dans les mémoires dynamiques ou dans les circuits à grande échelle d'intégration (LSI), le H-MOS s'emploie directement d'un bout à l'autre de l'éventail des produits.



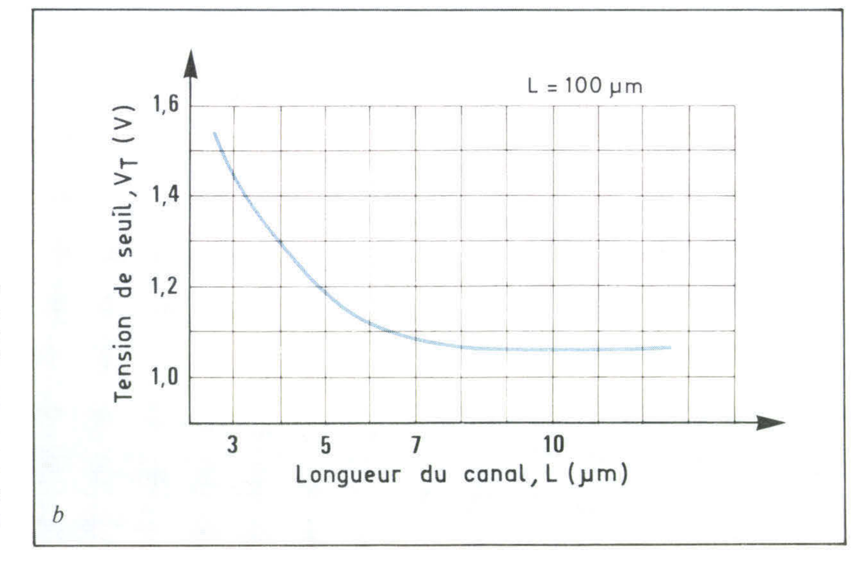
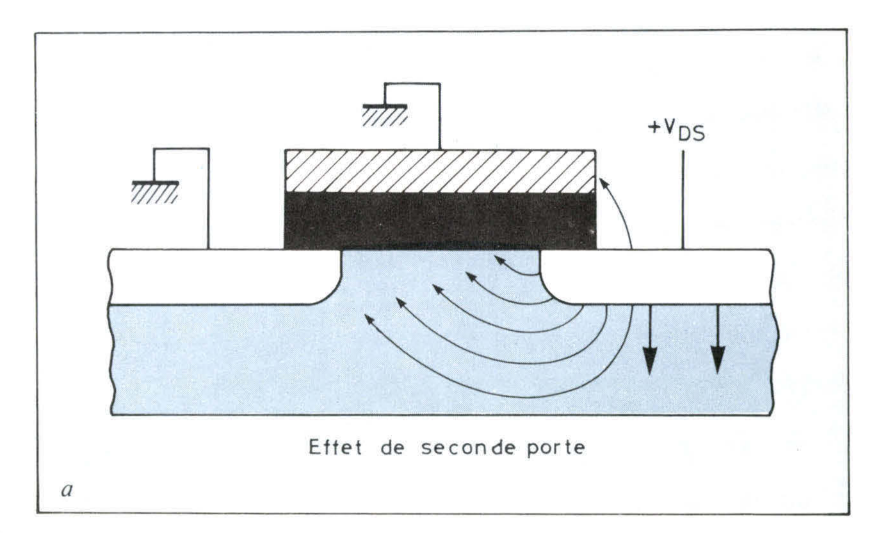


Fig. 4. – La réduction de la longueur et de la largeur du canal dans un élément MOS agit sur la tension de seuil. Lorsque la largeur du canal atteint 5 microns, V_T commence à décroître (a) ; tandis que pour des largeurs inférieures à 7 µm, V_T commence à monter (b).

Au-delà de la théorie du premier ordre

Alors que la taille des circuits diminue, la théorie prédit qu'ils



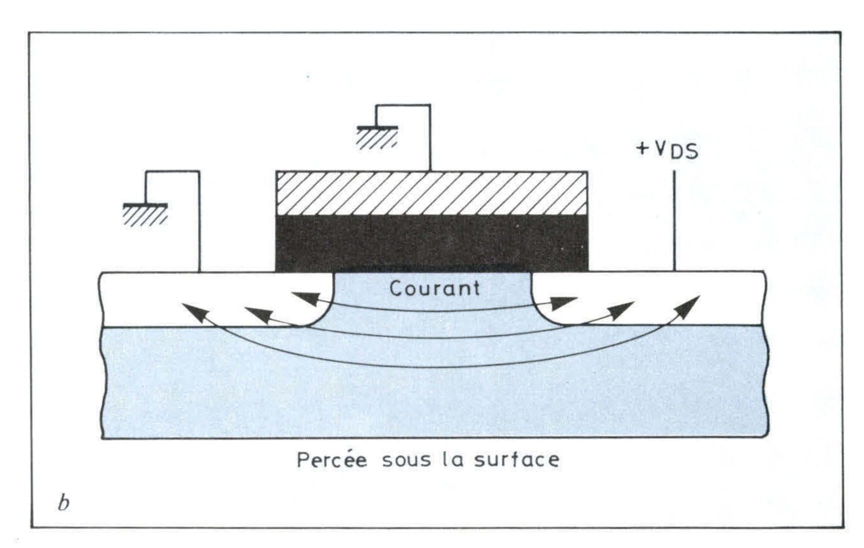


Fig. 5. – Les éléments de petite dimension sont vulnérables à deux effets secondaires. L'un est l'effet de seconde porte (a) où les lignes de force du champ électrique émanant de la jonction drain aboutissent à la surface de séparation oxyde-silicium. L'autre est le courant dû à la percée sous la surface (b) à laquelle il peut être remédié par un choix soigneux du profil d'impureté du substrat, réalisé par implantation ionique, combinée à une couche mince d'oxyde.

doivent conserver les mêmes caractéristiques qualitatives. Mais, dans la réalité, les phénomènes du second ordre deviennent non négligeables. Certains de ces phénomènes affectent le dessin du circuit, tandis que d'autres ont une incidence sur la fiabilité, mais tous doivent être pris en considération et compris parfaitement si l'on veut s'assurer que le H-MOS est un processus utile et sûr. Tous les effets du second ordre se produisent essentiellement pour deux raisons. La première est que, les dimensions étant réduites alors qu'une tension d'alimentation constante (5 V) reste appliquée, le champ électrique moyen augmente et crée de nombreux effets secondaires. La seconde est que les bords d'un petit élément sont si rapprochés que des champs électriques se produisent en ces bords et affectent les performances de l'élément d'une façon significative.

L'un de ces effets secondaires affecte la tension de seuil qui devient une fonction notable de la taille de l'élément. Comme le montre la **figure 4a,** pour les canaux les plus courts (ceux de moins de 5 μm de long), la zone de déplétion de jonction autour de la source et du drain tend à absorber une partie de la charge d'impuretés ionisées que la tension de porte aurait autrement maintenue, réduisant ainsi la tension de seuil. D'autre part, pour les canaux les plus étroits **(fig. 4b),** la charge ionisée supplémentaire créée par les champs électriques tend à augmenter la tension de seuil.

En outre, dans les circuits à canal court, les caractéristiques d'isolation entre source et drain peuvent aussi affecter le fonctionnement. La porte et la source d'un élément à enrichissement étant à la masse, le drain doit être capable de supporter une certaine tension positive et continuer à maintenir le courant de fuite à un certain niveau nominal. Dans les transistors à haute performance, ce courant de fuite peut avoir plusieurs causes, mais pour le H-MOS, le facteur de limitation est le phénomène appelé percée de la seconde porte **(fig. 5).**

Dans ce phénomène, les lignes de force du champ électrique émanant de la jonction drain aboutissent à la surface de séparation oxyde-silicium du canal. Le drain se comporte ici comme une seconde porte parasite et provoque l'inversion du canal, rendant ainsi l'élément plus vulnérable aux effets de percée.

Le problème, pour les éléments à canal court, est que la tension de percée qui résulte de cet effet est une fonction linéaire de la longueur du canal **(fig. 6).** Plus court est le canal, plus basse est la tension qui provoque la percée et par conséquent plus vulnérable est l'élément vis-à-vis des fuites. Dans le H-MOS, on remédie cependant à cela en gardant une longueur de canal suffisante et en réduisant l'épaisseur d'oxyde, puisqu'une couche d'oxyde plus mince empêche les inversions indésirables en couplant, par capacité, le potentiel de surface plus étroitement à l'électrode de porte mise à la masse. Un second effet de percée se produit, comme le montre la **figure 4b,** quand le champ électrique issu du drain atteint la source à travers le substrat et polarise la jonction, provoquant un courant de fuite (semblable à ce qui se passe dans un transistor bipolaire) mais ici encore, cette tension de claquage, proportionnelle à L^2, n'est un facteur de limitation que pour les éléments plus petits que ceux employés actuellement dans les structures H-MOS.

Ionisation d'impact

Une autre cause de fuite est l'ionisation d'impact dont les effets

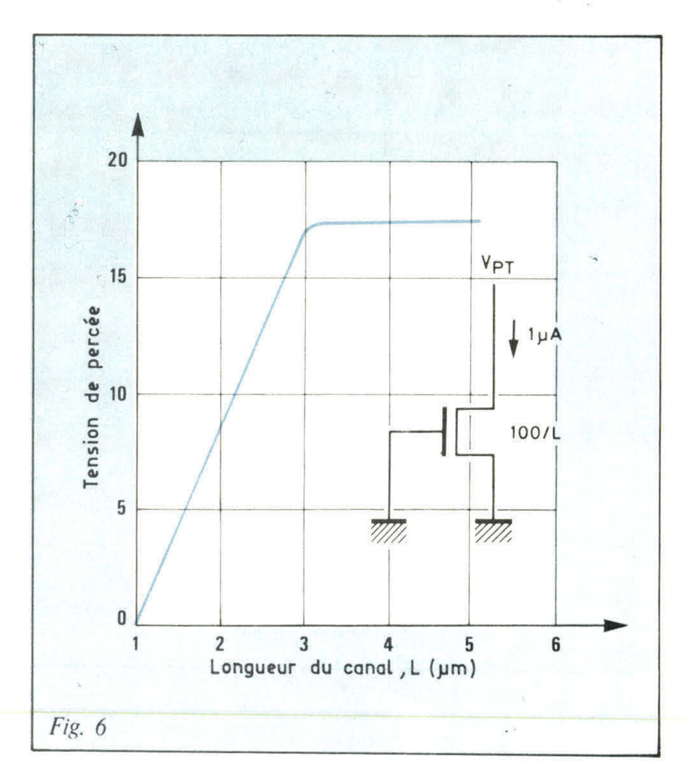


Fig. 6

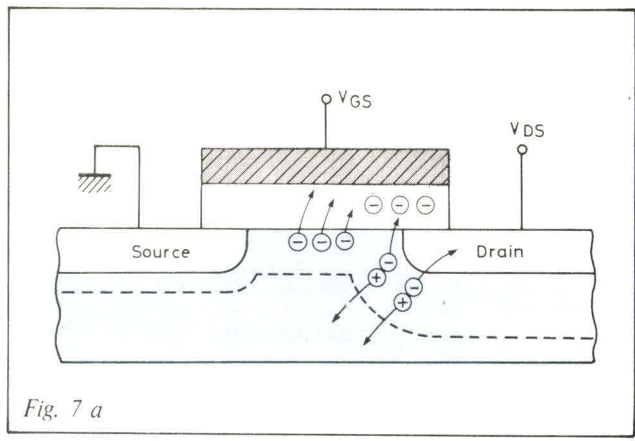


Fig. 7 a

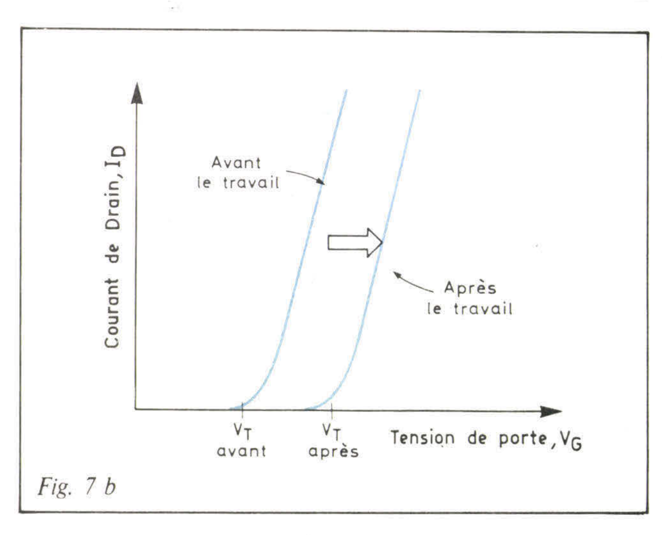


Fig. 7 b

sont illustrés à la **figure 7a.** A une tension de drain très élevée (de l'ordre de 20 V) les jonctions entrent en avalanche pour toutes les longueurs de canal plus grandes que 4 μm environ. Mais même à des tensions de drain H-MOS sensiblement plus faibles (5 V), une faible ionisation d'impact peut se produire lorsque le courant traverse le canal. Activée par les champs électriques intenses, l'ionisation d'impact crée une population d'électrons et de trous avec des énergies beaucoup plus élevées que celle des électrons normaux du canal. Les trous s'écoulent dans le substrat et placent une légère charge sur l'alimentation. Comme le montre la figure, quelques électrons ont assez d'énergie pour être injectés dans l'oxyde de porte, où ils peuvent créer un courant de porte ou bien rester piégés. Ces électrons piégés provoquent un déplacement de la tension de seuil **(fig. 7b)** – déplacement qui pourrait poser un problème avec des canaux de moins de 4 μm de long.

Heureusement, il s'avère que le problème de fiabilité qui pourrait se poser et que la théorie du second ordre a mis en évidence (à savoir le piégeage des électrons injectés dans l'oxyde de porte) n'affecte pas le H-MOS dans sa forme actuelle. Bien sûr, les électrons piégés tendent à accroître la tension de seuil, et un accroissement de la tension de seuil peut dégrader la vitesse du circuit ou arrêter complètement son fonctionnement. Mais des tests accélérés sur des circuits de mémoire H-MOS ne révèlent aucun signe de dégradation.

Fig. 6. – Dans les éléments MOS à canal court, la tension de percée atteint des niveaux qui peuvent causer des problèmes de fuite. La solution est de garder la longueur de canal dans la zone des 4 μm, et de veiller à ce que l'oxyde de porte soit mince.

Fig. 7. – La charge piégée dans l'oxyde de porte des éléments à canal court peut être si importante (a) qu'elle peut conduire à un déplacement permanent de la tension de seuil (b). Ce déplacement peut poser un problème de fiabilité dans ces éléments, mais on peut le réduire au minimum par un traitement très soigneux de l'oxyde et par une réduction de la tension d'alimentation.

En fait, des mesures supplémentaires sur les transistors individuels, complétées par un modèle physique pour l'injection d'électrons, montrent que les éléments H-MOS ont un déplacement de seuil total de moins de 0,1 V après 10 ans de travail dans les conditions de sévérité limite (à 0 °C et avec un V_{DS} de 5,5 V). Le traitement soigneux de l'oxyde de porte rend compte partiellement de ces résultats excellents, car il minimise la densité des zones de piégeage des électrons. De plus, il est intéressant de noter que les oxydes de porte de ces H-MOS à 5 V sont soumis à un effort moindre que les oxydes des mémoires RAM dynamiques à 12 V actuelles, car 5 V à travers une épaisseur d'oxyde de 700 angströms, cela fait un champ électrique plus faible que 12 V à travers l'épaisseur d'oxyde de 1000 angströms standard.

Une condition essentielle doit être remplie afin que la réduction d'échelle des éléments dans la technologie H-MOS puisse réaliser pleinement ses possibilités.

Cette condition est la réduction de la tension d'alimentation. Le **tableau 2** montre que, pour donner à cette technique une chance de fonctionner, les tensions d'alimentation des circuits à 2 μm de l'année 1980 doivent être ramenées dans la zone de 2 à 4 V. Etant donné que toutes les tensions d'alimentation sont maintenues actuellement au moins au niveau TTL de 5 V, cette exigence d'une tension d'alimentation plus faible pour les futurs éléments MOS doit être acceptée par les utilisateurs de circuits intégrés.

Il y a, bien sûr, une autre possibilité pour les utilisateurs qui refusent tout simplement d'accepter des tensions d'alimentation réduites. Ils peuvent utiliser des circuits adaptateurs pour effectuer la conversion entre les niveaux de tension inférieurs des puces et les niveaux supérieurs des entrées/sorties TTL, ou bien, ils peuvent employer deux sources d'alimentation, l'une fournissant 2 à 4 V aux circuits internes de la puce, et l'autre fournissant 5 V à

leurs circuits d'entrée/sortie. Mais l'une ou l'autre méthode est au mieux un palliatif.

La clé de l'avenir : une tension d'alimentation réduite

Une tension d'alimentation plus faible augmente la fiabilité des éléments de dimensions réduites, tout en accroissant en même temps leurs performances d'une façon remarquable. La fiabilité augmente parce qu'une tension d'alimentation plus basse entraîne une plus grande tolérance vis-à-vis de la tension de percée, et produit en même temps des champs électriques plus faibles dans la région du canal. Ce second effet réduit le risque que des charges soient piégées dans l'oxyde de porte ce qui altère la stabilité à long terme de l'élément.

L'accroissement en performance est encore plus frappant. Pour les mémoires vives et les microprocesseurs **(fig. 8)**, l'incidence sur le produit vitesse-puissance suffirait à elle seule à justifier le passage d'une tension d'alimentation de 5 V à 3 V. Pour les mémoires vives qui travaillent au point de saturation de la courbe vitesse-puissance, la tension d'alimentation plus faible réduit la puissance dissipée d'environ 60 % tout en maintenant la vitesse à la même valeur élevée. Cette réduction de la puissance dissipée devient extrêmement importante lorsque la densité de la puce croît (à 65 536 bits et à 262 144 bits) car il est généralement admis que pour avoir un fonctionnement fiable, la puissance dissipée pour chaque boîtier doit être maintenue en-dessous du watt.

En ce qui concerne les microprocesseurs, comme ils travaillent dans une région bien éloignée du point de saturation de la courbe vitesse-puissance, ils peuvent tirer pleinement profit de l'accroissement significatif de vitesse pour une puissance dissipée donnée. Une puce de microprocesseur à 3 V opérera par exemple à deux fois la vitesse d'un élément alimenté en 5 V.

La façon dont la tension d'alimentation affecte les différents processus MOS qui ont été développés au cours des années, est montrée à la **figure 9**, et de nouveau, les avantages des tensions réduites sont mis en évidence. En effet, pour la version 1980 des éléments MOS (longueur de canal de 2 μm et épaisseur d'oxyde de 400 angstroms) une alimentation de 2 à 4 V donnera un délai de porte deux fois moindre que celui du processus H-MOS actuel qui opère à 5 V.

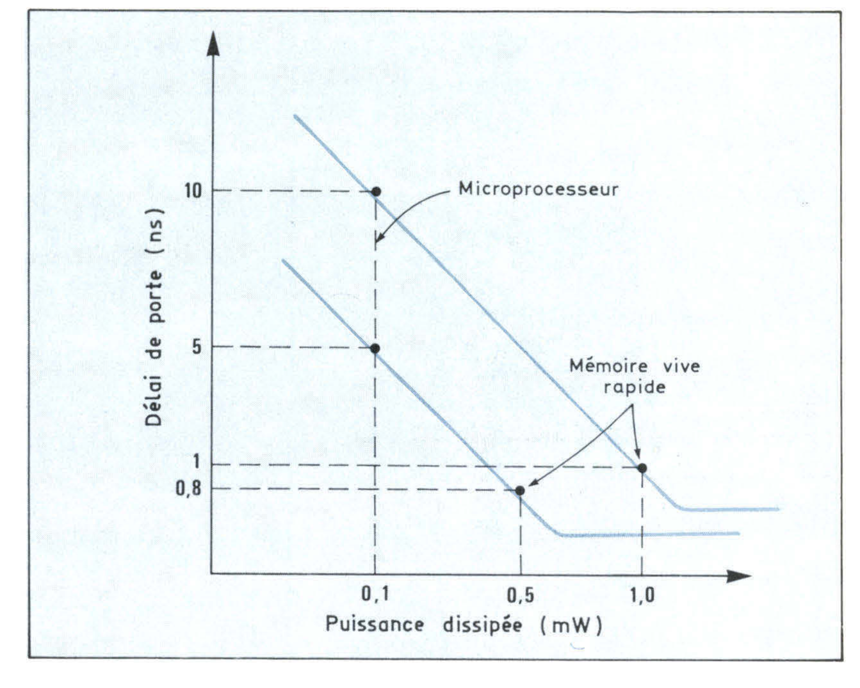


Fig. 8. – En réduisant la tension d'alimentation de 5 à 3 V, on améliore de façon notable la vitesse et la puissance dissipée des circuits MOS, plus particulièrement dans les éléments réduits. L'effet est plus important pour les microprocesseurs, où une réduction de 2 V dans la tension d'alimentation conduit à un doublement des performances.

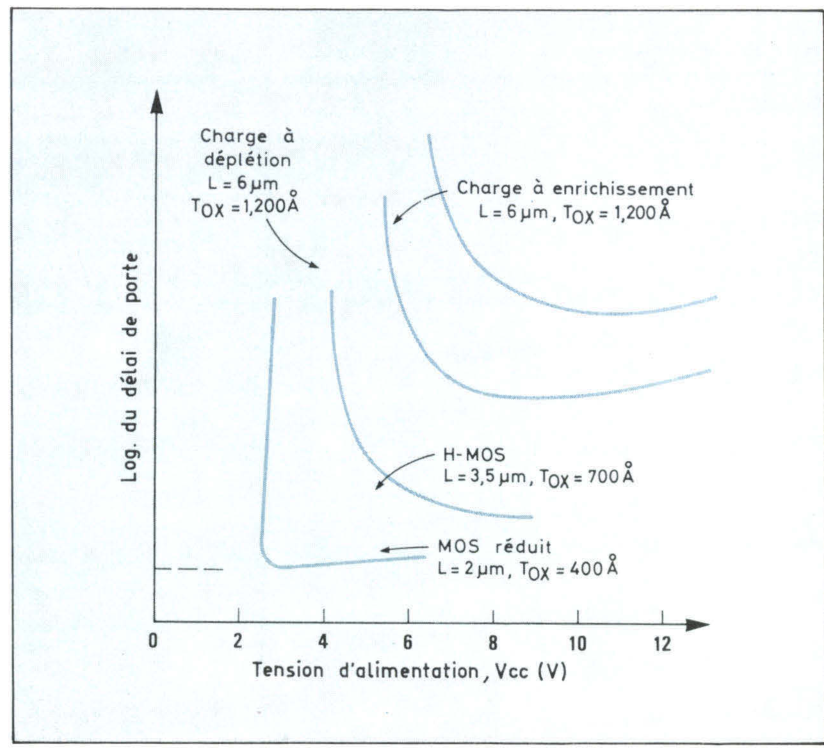


Fig. 9. – A mesure que la technologie MOS s'améliore et que les éléments deviennent plus petits, le besoin d'une tension d'alimentation réduite devient plus urgent. Pour des longueurs de canal de 2 microns, une alimentation de 3 V donne le meilleur délai de porte et une plus grande fiabilité du système.

Principaux avantages des H MOS

Trois approches possibles permettent la réalisation de MOS aux hautes performances, ce sont les H.MOS, les MOS à sillon en V, les SOS (Silicium sur saphir ou Silicon On Sapphire). Comme le montre le tableau ci-dessous, les versions actuelles des H.MOS et des V.MOS donnent toutes un produit vitesse-puissance d'environ 1 picojoule.

Le **V.MOS** offre, en principe, une meilleure densité d'intégration, mais au prix d'un processus de fabrication plus complexe. En outre, le V.MOS produit un circuit asymétrique qui ne doit être utilisé que dans un seul sens, de sorte que les configurations logiques d'intégration à grande échelle (LSI) sont plus difficiles à réaliser qu'avec les H.MOS.

Le **SOS,** quant à lui, présente le meilleur produit vitesse-puissance. Mais il nécessite un substrat cinq à sept fois plus cher et ne semble se justifier que pour les applications microprocesseur qui n'exigent pas le fonctionnement dans la zone de haute vitesse de la courbe vitesse-puissance.

Le principal avantage que les H.MOS possèdent actuellement sur les V.MOS réside dans le fait qu'ils peuvent être directement développés par réduction d'échelle à de plus hautes performances et à une plus grande densité et ceci à un plus faible coût. Les performances des MOS réduits de 1980 (canaux de 2 microns) sont indiquées ci-dessous. Elles sont environ cinq fois meilleures que dans la technologie actuelle.

COMPARAISON ENTRE TROIS TECHNOLOGIES MOS

Paramètres	H.MOS 1977	H.MOS réduit 1980+	V.MOS 1977	SOS 1977
Densité d'intégration (portes/mm^2)	170	200	220	150
Produit vitesse-puissance (pJ)	1	0,2	1	0,2
Délai de porte (nS)	1	0,4	1	0,5
Nombre de films minces	2	2	3	3
Nombre d'implantations	3	3	3	2

Les trois niveaux de langage

La tendance actuelle de l'évolution de l'électronique est sans conteste une régression des informations traitées en analogique au profit des informations traitées du point de vue logique. D'ici peu, dans une chaîne analogique, en dehors de la source et de l'utilisation, la majorité des problèmes sera résolue par des solutions en circuits logiques, encadrés, bien entendu, par des convertisseurs analogique/digital et digital/analogique.

L'implantation des circuits logiques fait ensuite penser aux solutions programmées : de petits microprocesseurs « toutes utilisations », des mémoires et du logiciel. Si bien que l'on peut prévoir sans grands risques d'erreur que l'évolution analogique — logique et logique câblée — programmée conduira, quel que soit le type de problème, à choisir un microprocesseur dans un lot de composants standards et à associer de la mémoire et du logiciel

L'intégration des circuits va se poursuivre, ils seront de plus en plus performants, de faibles dimensions et de prix réduit. Par contre, pour suivre cette évolution, les programmes devront être plus sophistiqués, subtils et imbriqués afin que ce qui est gagné par le matériel ne soit pas gaspillé en logiciel.

L'informatique est venue de l'électronique sans laquelle elle n'existerait pas. Et maintenant, grâce aux progrès technologiques de l'intégration, ces deux techniques se rapprochent pour s'interpénétrer de plus en plus. L'électronicien ne disparaît pas mais se transforme en programmant maintenant le matériel qu'il utilise. Après tout, cette évolution n'est-elle pas l'essence même de cette science qui a connu en si peu de temps : tubes, transistors, circuits intégrés, microprocesseurs, alors pourquoi pas le langage...

	0100	00010110	16
	0101	—	γ
	0102	00001110	0E
	0103	11111111	FF
	0104	00000110	06
	0105	11111111	FF
	0106	00000101	05
	0107	11000010	C2
	0108	00000110	06
	0109	00000000	00
	010A	00001101	0D
	010B	11000010	C2
	010C	00000100	04
	010D	00000000	00
	010E	00010110	16
	010F	11000010	C2
	0110	00000010	02
a	0111	00000000	00

	MVI	D,γ	D est chargé à la valeur γ
B1	MVI	C,FF	C est chargé à FF
B2	MVI	B,FF	B est chargé à FF
B3	DCR	B	Décrémentation de B de 1
	JNZ	B3	Si B $\neq$ 0 (pas zéro) aller en B3
	DCR	C	Décrémentation de C de 1
	JNZ	B2	Si C $\neq$ 0 aller en B2
	DCR	D	Décrémentation de D de 1
b	JNZ	B1	Si D $\neq$ 0 aller en B1

```
10 REM TEMPO A UNE BOUCLE
20 INPUT D
30 FOR I = 1 TO 10 000 * D STEP 1
40 NEXT I
50 END
```
c

Le même programme (fonction de temporisation) écrit en langage machine (a), en langage assembleur (b) et en langage évolué (c).

** Machine sera pris au sens le plus large, c'est-à-dire du plus petit des microprocesseurs au plus gros des ordinateurs.*

Le langage est un code qui permet de converser avec la machine *.

L'ensemble des langages existants peut se diviser en trois familles ou plus exactement trois niveaux d'utilisation.

- **Niveau 0** ou langage machine. Il est à base de « 0 » et de « 1 » et correspond au fonctionnement réel du système.

- **Niveau 1** ou langage assembleur, un peu semblable au précédent mais utilise un code mnémonique, abréviation anglo-saxonne de l'instruction à exécuter.

- **Niveau 2** ou langage évolué, c'est un langage de haut niveau qui fait totalement abstraction de l'architecture de la machine utilisée. Il se rapproche beaucoup du langage courant.

Rappels préliminaires

On appelle BIT l'unité élémentaire d'information. C'est la contraction de :

Binary digIT

Dans un circuit électrique la tension est appliquée ou non. Si elle l'est, on dit que l'on est au niveau

1, sinon c'est le niveau 0. Le bit correspondant aura ainsi pour valeur 0 ou 1. C'est un langage binaire. L'association de huit bits formera un Octet (ou byte en anglais). Suivant leur conception, les machines courantes seront à 4, 8, 16, 32 ou 64 bits. Ce sera la taille de la mémoire élémentaire appelée aussi mot mémoire.

Actuellement, car l'évolution est très rapide, les microprocesseurs sont principalement à 8 ou 16 bits alors que les ordinateurs possèdent des mots mémoires plus longs : 32 ou 64 bits.

Un registre est un emplacement mémoire personnalisé et privilégié. Il est en général repéré par une lettre et a souvent la même taille que le mot mémoire.

Une machine travaille uniquement en binaire. L'emploi d'une autre notation : octal, hexadécimal, ... n'est qu'une représentation commode du binaire.

Pourquoi l'hexadécimal ? Si l'on considère un demi-octet, les quatre bits pourront représenter.

$2^4 = 16$ combinaisons

Pour la notation de 0 à 9 il n'y aura pas de difficulté, l'équivalence sera totale. Ensuite, il faudra de nouveaux symboles. Les lettres de A à F ont été choisies ce qui nous donne le tableau suivant :

décimal	binaire	hexadécimal
1	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	2
3	0 0 1 1	3
4	0 1 0 0	4
5	0 1 0 1	5
6	0 1 1 0	6
7	0 1 1 1	7
8	1 0 0 0	8
9	1 0 0 1	9
10	1 0 1 0	A
11	1 0 1 1	B
12	1 1 0 0	C
13	1 1 0 1	D
14	1 1 1 0	E
15	1 1 1 1	F

Après F, qui est aussi 0F, on procède comme en décimal. Lorsque le cycle est terminé on augmente les dizaines de une unité et l'on recommence.

Après 29 on dira 30, pour l'hexadécimal, après 0F on aura 10, puis 11, etc., qui, en réalité, ne doivent pas s'énoncer dix, onze, etc., mais : un zéro, un un,...

Pour éviter la confusion de systèmes entre décimal et hexadécimal, on emploie dans ce dernier cas le signe H placé après le nombre (ex : 10H).

L'organisation d'une machine sera, par exemple, celle représentée sur la **figure 1,** et qui correspond au microprocesseur 8080.

Chaque rectangle élémentaire a pour dimension un octet. L'ensemble est formé de plusieurs registres repérés par les lettres : A, B, C, D, E, H et L. Le registre A ou accumulateur est privilégié dans les opérations et lui seul peut communiquer avec l'extérieur.

La mémoire extérieure associée au microprocesseur est formée d'une succession d'octets repérés par leur adresse. Toutes ces notations sont écrites en hexadécimal.

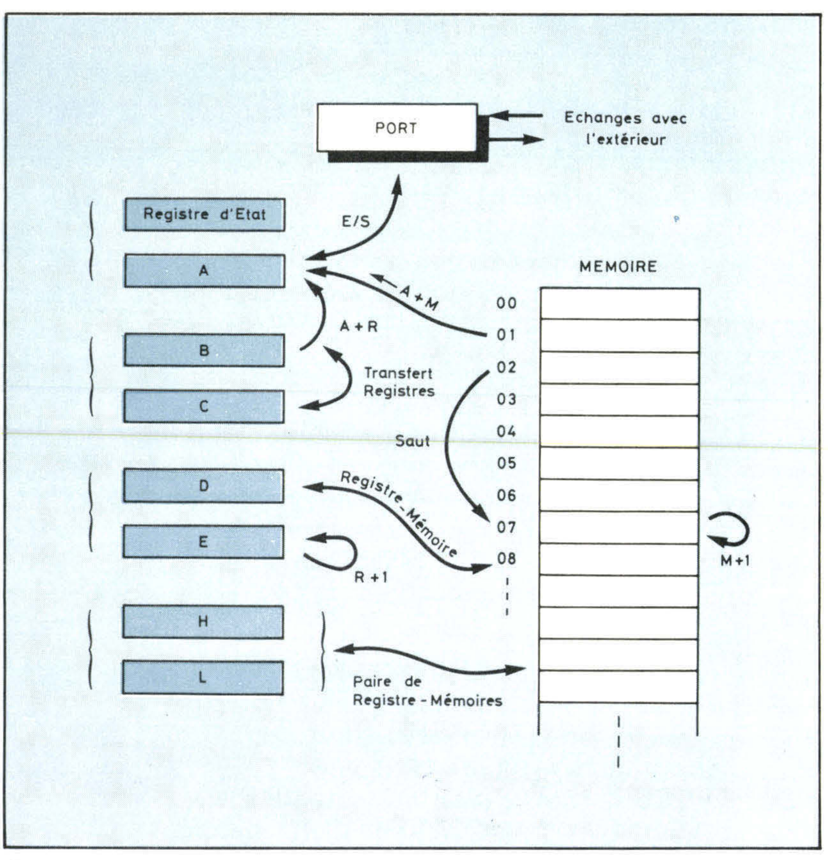


Fig. 1. – Structure générale des registres du 8080 avec une mémoire associée. Les principaux échanges possibles sont représentés par les flèches.

Il serait assez rébarbatif d'apprendre d'emblée toutes les instructions d'un microprocesseur donné. Il est beaucoup plus astucieux par contre de savoir quels sont les genres d'opérations existants.

Les principales familles sont les suivantes :

- transfert de données entre registres ou entre mémoire et registre ;
- groupe arithmétique : addition et soustraction par rapport à l'accumulateur, incrémentation et décrémentation des contenus de registre ou de mémoire ;

- groupe logique : réalisation de fonction OU, OU exclusif, ET entre le contenu de l'accumulateur et celui d'un registre ou d'une mémoire, comparaisons, complémentation de données, rotations dans les registres ou la mémoire ;

- les sauts : conditionnels ou inconditionnels appel de sous-programmes.

Cette dernière famille gère, en dehors des fonctions d'entrées-sorties, le fonctionnement de la machine (pile, indicateurs d'états).

Au point de vue du fonctionnement ; dans une zone mémoire sera situé en binaire le programme à exécuter, dans une autre seront stockés les données et les résultats. Un pointeur (ou registre) indiquera l'adresse de l'instruction à exécuter.

Programmer en langage machine consiste donc tout simplement à entrer à l'aide de clés ou d'un clavier hexadécimal les instructions en langage machine. C'est un travail long, fastidieux, et source d'erreurs.

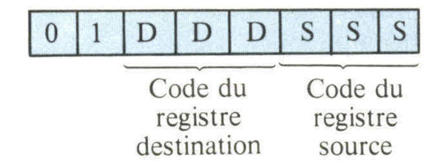
Les instructions en langage machine

Suivant le type elles sont stockées dans un, deux ou trois octets.

• Exemple d'instruction codée sur 1 octet

Recopie d'un registre dans un autre registre :

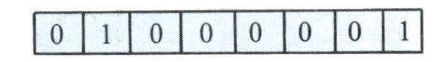
Code opération



Si l'on recopie le registre C (source) dans le registre B (destination) dont les codes sont par exemple :

Code B : 0 0 0
Code C : 0 0 1

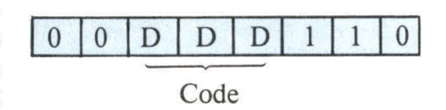
L'instruction en langage machine sera représentée par le symbole (B)←(C) et son code sera :



• Exemple d'instruction codée sur 2 octets

Chargement d'un registre :

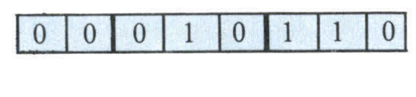
Code opération

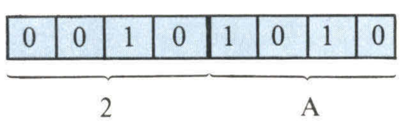


donnée

donnée ou adresse

Soit à charger le registre D dont le numéro de code est : 010 par la valeur 2AH, nous aurons :



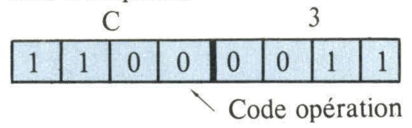


Ce qui s'écrit symboliquement de la façon : (D)←2AH.

• Exemple d'instruction codée sur 3 octets

Saut en une adresse déterminée :

C'est l'équivalent de **GO TO** ou ALLER EN mais il s'agit dans ce cas d'adresse réelle en mémoire et non d'étiquette.

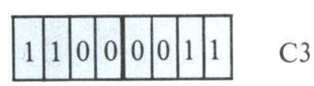


partie basse de l'adresse	donnée ou adresse
partie haute de l'adresse	

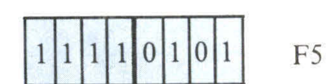
L'adresse étant codée sur quatre chiffres hexadécimaux ce qui permet de couvrir 65 536 cases mémoires différentes, soit $(16)^4$. C'est le nombre maximum de mémoire que peut adresser le système, encore faut-il qu'elles soient physiquement présentes. Par convention au niveau du fabricant la partie basse de l'adresse c'est-à-dire les chiffres les moins significatifs sont stockés dans le deuxième octet et la partie haute dans le troisième.

Soit un saut à effectuer à l'adresse mémoire 01F5H.

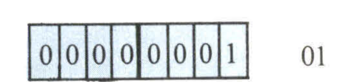
Code opération



C3



F5



01

Cette instruction charge la valeur 01F5H dans le compteur de programme et se note :

(PC)←01F5 H

Le langage assembleur

Le langage assembleur est plus commode que le précédent car des expressions mnémoniques (bien qu'en anglais) Call, Load, Move..., sont quand même d'un emploi plus commode que des combinaisons de « 0 » et « 1 ». Le programme en assembleur se présentera en général sous forme de quatre colonnes :

Il ne faut pas confondre le programme écrit en langage assembleur avec le programme assembleur qui, lui, est le traducteur chargé de transformer le premier en langage machine.

Label (étiquette)	Instruction	Opérande	Commentaire
	MVI	B, K	le registre B est chargé par K
B	INR	C	le contenu du registre s'est augmenté de 1

Les **LABELS** sont des repères pour les boucles. Dans l'écriture en assembleur, nous ne nous soucions plus de l'emplacement en mémoire et en réalité nous ignorons où se trouvent les instructions. Par exemple seule la machine sait où se trouve l'instruction INR, pour nous il nous suffit de savoir qu'elle se trouve à l'étiquette B. Certaines instructions ne nécessitent pas de repère c'est pour cela que les lignes correspondantes de cette colonne peuvent être vides.

Les **« INSTRUCTIONS »** sont représentées sous forme codée, soit par exemple :

MVI : MOVE IMMEDIATE
(chargement immédiat
d'un registre)

INR : INCREMENT REGISTER
(augmentation du contenu d'un registre de 1 unité)

La colonne **« OPERANDE »** précise le registre, la donnée ou l'adresse liée à l'instruction. C'est-à-dire par exemple quel est le registre dont le contenu sera augmenté de un, etc.

Les **« COMMENTAIRES »** ne sont pas indispensables mais facilitent la compréhension, vérification ou modification d'un programme. Avec un peu d'expérience, une forme assez condensée pourra être utilisée.

De par son principe même, le langage assembleur semble d'un emploi plus commode que le langage machine. Il sera entré à l'aide d'un clavier.

Un dernier point reste à résoudre : qui va assurer la liaison entre les deux langages ? Ce sera un programme de traduction, résidant en mémoire et qui s'appellera programme d'assemblage, programme assembleur ou tout simplement assembleur.

Il ne faudra évidemment pas confondre le programme écrit en langage assembleur avec le programme assembleur (traducteur) qui sera lui chargé de transformer le premier en langage machine pour exécution.

Le langage évolué

A quelques subtilités près concernant l'écriture mathématique, ou des instructions de sortie ce langage est très compréhensible. Il peut être parfois plus ou moins spécialisé vers la gestion ou le calcul scientifique.

En 1976 aux Etats-Unis les pourcentages d'utilisation des principaux langages s'établissaient comme suit :

FORTRAN : 47 %
BASIC : 27 %
COBOL : 13 %
APL : 5 % (mais en très forte progression)
DIVERS : 8 %

Bien que ces différents langages soient d'origine anglo-saxonne donc utilisant des instructions dans cette langue, signalons un langage français : le LSE (langage symbolique d'enseignement) assez similaire au Basic mais plus performant.

Comme dans le cas du langage assembleur, un programme interne beaucoup plus important que le précédent résidera dans la machine où il occupera une importante zone mémoire. Ce programme traducteur se nommera dans ce cas Compilateur.

Par exemple un compilateur Fortran assurera la liaison entre le programme en Fortran entré par la télétype et le langage machine nécessaire au fonctionnement.

Exemple de programme

Pour bien mettre en évidence les trois types de langages, nous reproduisons ci-après le même programme écrit en langage machine, en langage assembleur et en langage évolué.

L'exemple choisi est une fonction de temporisation. Ces fonctions sont très utiles en automatisme car dans de nombreuses applications sont associés des éléments mécaniques auxquels il faut toujours une certaine constante de temps entre l'ordre (application d'une tension) et la fin de l'exécution (fermeture d'un relais par exemple).

L'organigramme est reproduit **figure 2.** L'équivalent mécanique du principe de fonctionnement est tout simplement un système d'engrenage formé de trois roues dentées, chacune d'elle tourne d'un cran lorsque la précédente a fait un tour complet. Notons, en passant, que très souvent la transposition informatique d'un système simple apporte la solution cherchée.

Analyse du fonctionnement :

Les registres B et C sont chargés à leur capacité maximale : tous les bits à 1 soit FF.

Le contenu du registre D sera ajusté en fonction du temps de temporisation souhaité.

Le contenu de B est diminué de une unité, on teste si son contenu est nul, retour à la boucle 3, nouvelle diminution de un, etc.,

jusqu'à ce que le contenu de B soit nul. A ce moment, qui correspondrait à un tour complet de la première roue dentée, c'est le registre C qui est diminué de un, puis test et retour éventuel de la boucle 2 où B est une nouvelle fois chargé à sa capacité maximale et devra se vider à nouveau complètement.

Lorsque B, C et D sont vides, la temporisation est terminée et la suite du programme peut se dérouler normalement.

Dans le cas du microprocesseur 8080 la durée de temporisation c'est-à-dire la durée d'exécution du programme est donnée par la relation :

$$\tau = 0{,}621 \times \gamma$$

τ en secondes
γ contenu du registre D en décimal.

Les programmes dans les trois langages sont reproduits aux **figures 3, 4** et **5.**

Fig. 3. – Listing du programme en langage machine. Le programme est implanté à partir de l'adresse 100. Les adresses mémoire sont aussi notées en hexadécimal, γ est le contenu du registre D.

Adresse Mémoire	Contenu mémoire	
	Binaire	Héxadécimal
0 1 0 0	0 0 0 1 0 1 1 0	1 6
0 1 0 1	–	γ
0 1 0 2	0 0 0 0 1 1 1 0	0 E
0 1 0 3	1 1 1 1 1 1 1 1	F F
0 1 0 4	0 0 0 0 0 1 1 0	0 6
0 1 0 5	1 1 1 1 1 1 1 1	F F
0 1 0 6	0 0 0 0 0 1 0 1	0 5
0 1 0 7	1 1 0 0 0 0 1 0	C 2
0 1 0 8	0 0 0 0 0 1 1 0	0 6
0 1 0 9	0 0 0 0 0 0 0 0	0 0
0 1 0 A	0 0 0 0 1 1 0 1	0 D
0 1 0 B	1 1 0 0 0 0 1 0	C 2
0 1 0 C	0 0 0 0 0 1 0 0	0 4
0 1 0 D	0 0 0 0 0 0 0 0	0 0
0 1 0 E	0 0 0 1 0 1 1 0	1 6
0 1 0 F	1 1 0 0 0 0 1 0	C 2
0 1 1 0	0 0 0 0 0 0 1 0	0 2
0 1 1 1	0 0 0 0 0 0 0 0	0 0

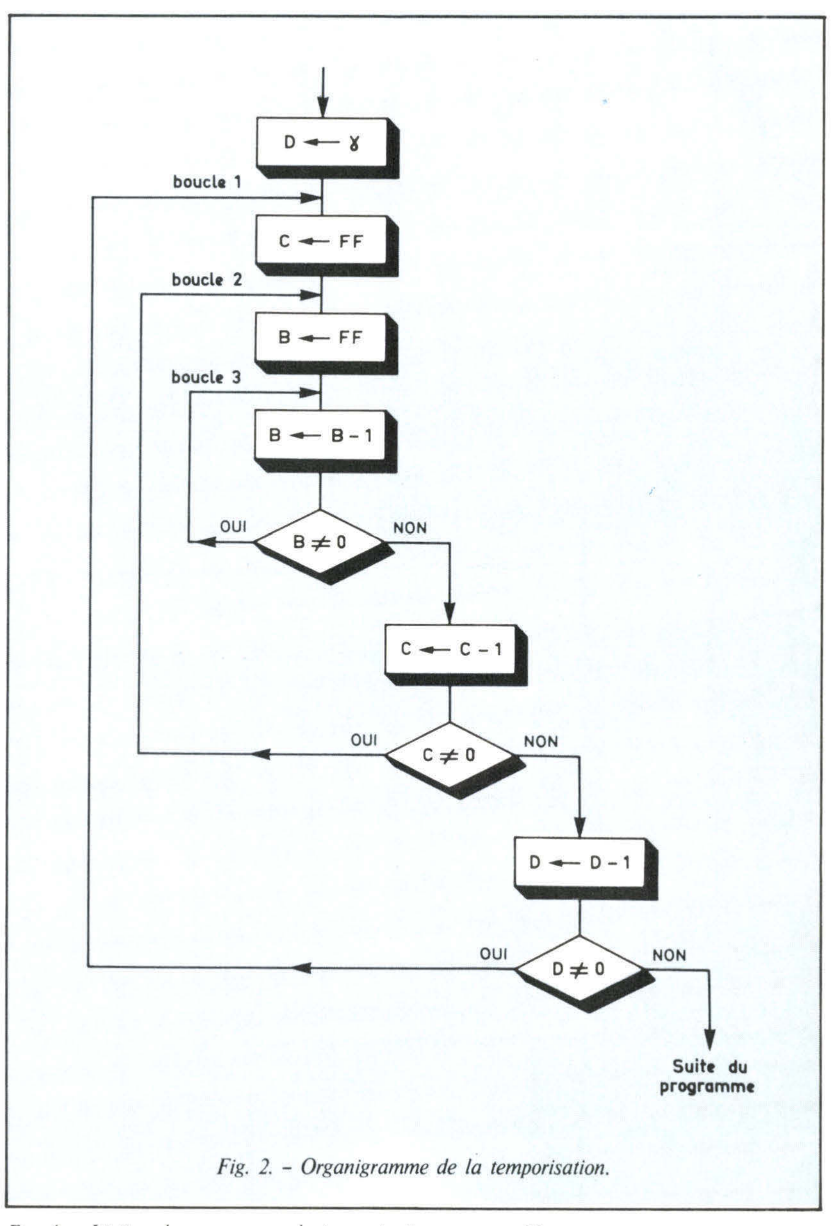


Fig. 2. – Organigramme de la temporisation.

Fig. 4. – Listing du programme de temporisation en assembleur.

Label	Instruc.	Opérande	Commentaire
	MVI	D,γ	D est chargé à la valeur γ
B1	MVI	C,FF	C est chargé à FF
B2	MVI	B,FF	B est chargé à FF
B3	DCR	B	Décrémentation de B de 1
	JNZ	B3	Si B ≠ 0 (pas zéro) aller en B3
	DCR	C	Décrémentation de C de 1
	JNZ	B2	Si C ≠ 0 aller en B2
	DCR	D	Décrémentation de D de 1
	JNZ	B1	Si D ≠ 0 aller en B1

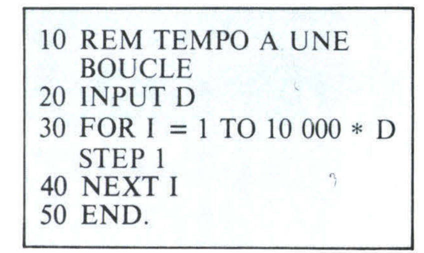
Même temporisation en langage évolué (BASIC).

Dans un premier temps, pour respecter l'organigramme à trois boucles on peut écrire le programme suivant **(fig. 5).**

```
10 REM TEMPO A TROIS
   BOUCLES
20 INPUT D
30 FOR I = 1 TO D STEP 1
40 FORK = 1 TO 10 STEP 1
50 FOR J = 1 TO 10 STEP 1
60 NEXT J
70 NEXT K
80 NEXT I
90 END.
```

Fig. 5
Listing du programme de temporisation en Basic.

Les limites supérieures dans les instructions FOR des lignes 40 et 50 ont été arbitrairement fixées à 10, pour faire ressembler le programme à sa variante en langage machine. Si le mot hexadécimal 8 bits permet d'écrire au maximum FF = 255_{10}, en basic on peut écrire une valeur quelconque (allant de 10^{-99} à 10^{+99} sur la plupart des machines) donc les trois boucles-roues dentées n'ont plus de sens. On peut écrire avantageusement :

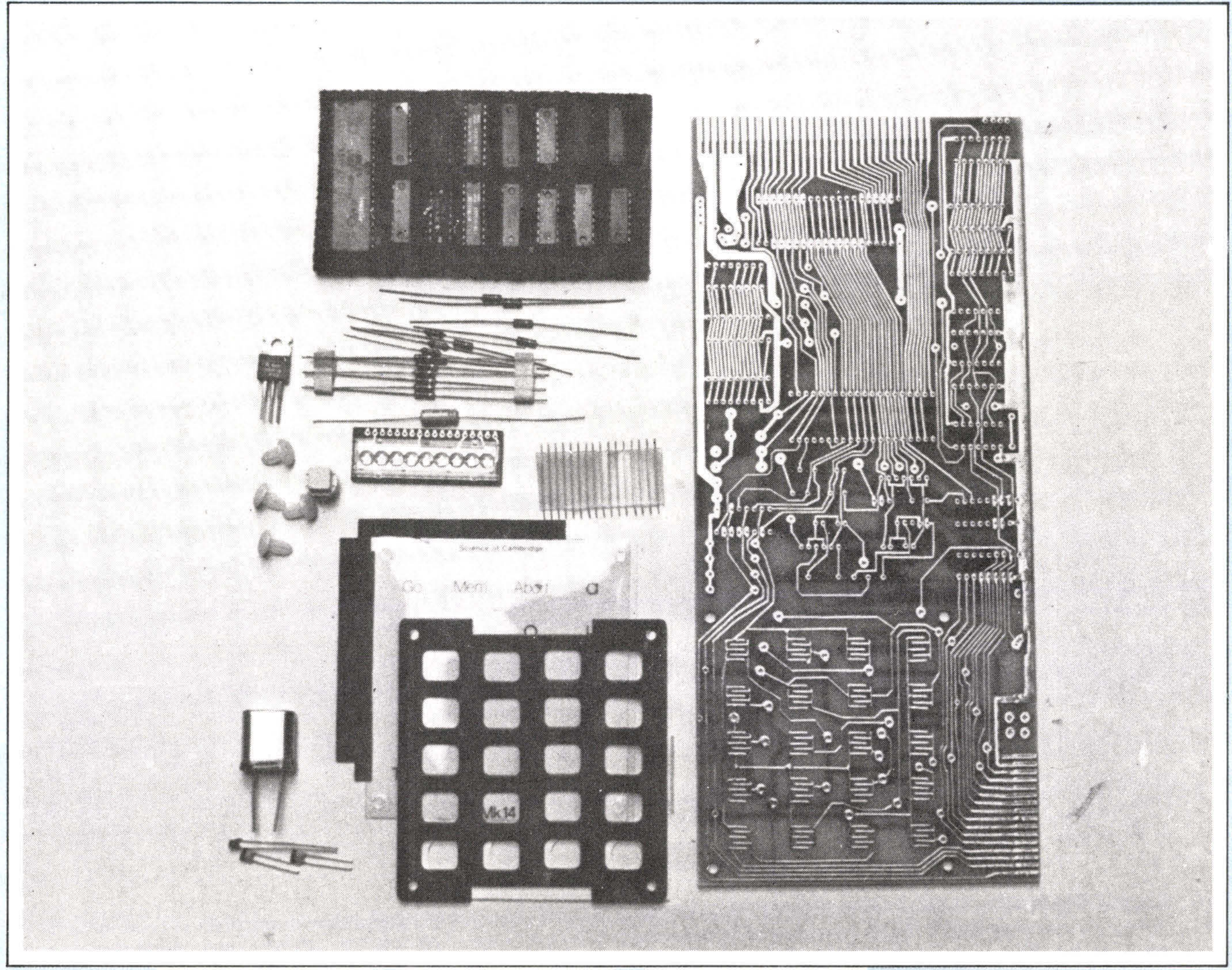


```
10 REM TEMPO A UNE
   BOUCLE
20 INPUT D
30 FOR I = 1 TO 10 000 * D
   STEP 1
40 NEXT I
50 END.
```

Après un premier passage par ce programme, l'utilisateur peut ajuster le coefficient qui fixe l'échelle du temps à sa guise. ■

André BILLÈS

MK 14



Le MK 14 : un kit d'initiation faible coût.

Le MK 14* a été commercialisé à la fin du premier semestre 1978, accompagné d'un manuel en anglais. J'ai eu la chance d'être l'un des premiers à être livré sur une longue liste d'attente, et à pouvoir recevoir la traduction française avant son impression. Celle-ci doit être disponible à l'heure actuelle.

Le MK14

Qu'est le MK 14 ? Un kit d'initiation aux microprocesseurs et à la micro-informatique. Il est conçu pour être d'un abord simple et attrayant, et d'un prix plus que raisonnable.

Cependant, ses extensions déjà annoncées, et celles qui suivront probablement, en font un appareil qui permet de poursuivre son utilisation bien au-delà de l'initiation.

Le MK 14 utilise le SC/MP II de National Semiconducteur, microprocesseur déjà abondamment décrit dans la presse spécialisée. Le micro-amateur peut donc se référer aux textes existants pour compléter éventuellement les explications du manuel.

Les composants sont mis en place sur un circuit imprimé double face de 25,5 x 11,5 cm, sur lequel viennent se fixer un clavier hexadécimal et un barreau afficheur à sept segments. L'appareil de base est livré avec une RAM de 256 octets et un moniteur de 512 octets en ROM. La mémoire peut être immédiatement portée à 640 octets de RAM, car les emplacements sont prévus.

Le manuel constitue l'un des attraits de ce système, et sa version française répond à beaucoup de questions que se sont posés — comme moi — les premiers acquéreurs.

Et quel encouragement lorsque, vingt minutes seulement après la mise sous tension, le premier « canard » du programme « Chasse au Canard » défilait sur l'afficheur !

Extensions possibles

En dehors de l'augmentation de mémoire RAM à 640 octets, un interface cassette économique — et certainement compatible avec l'Introkit — est disponible depuis quelque temps.

Mais, en questionnant le distributeur français, j'ai pu savoir qu'un programmeur de PROM devrait sortir sous peu, ainsi qu'un interface vidéo avec son modulateur UHF.

Le SC/MP peut, bien entendu, gérer plus de 640 octets de RAM, et il est possible, grâce à des circuits externes, d'augmenter facilement la mémoire jusqu'à 4 K octets.

Montage du kit

Le circuit imprimé reçoit 14 circuits intégrés. Les CMOS sont à monter sur supports pour éviter des soudures toujours délicates.

Dans ces conditions, le montage demande environ trois heures. Il est à la portée de personnes non-expérimentées — le schéma d'implantation est explicite, et le code de couleur est fourni.

Les instructions de montage, partie intégrante du manuel, consacrent même un chapitre spécial sur la manière d'effectuer de bonnes soudures ! Il est vrai que le MK 14 s'adresse en particulier aux futurs micro-informaticiens, qui peuvent ne pas être des artistes du fer à souder !

Le clavier porte la marque du constructeur, Science of Cambridge, qui s'est fait une spécialité de rechercher les solutions astucieuses. Une membrane souple, en caoutchouc conducteur, assure le contact entre les pistes du circuit imprimé.

Plusieurs membranes sont ainsi prises en sandwich entre le circuit imprimé et la plaque supérieure en métal. Mais j'ai préféré remplacer les rivets en plastiques qui doivent être insérés à force par de simples vis et écrous en nylon.

Remarquons cependant que le constructeur, prudent, a laissé la possibilité de se connecter à un clavier extérieur. Un connecteur à 16 broches est prévu à cet effet.

Afin de supporter la carte, j'ai simplement placé sous le clavier plusieurs pastilles auto-collantes de feutre, ce qui est d'ailleurs recommandé par le constructeur.

Mise en service

L'alimentation n'est pas exigeante : elle peut se faire entre 8 V et 25 V en courant redressé. Ces caractéristiques élastiques sont rendues possibles grâce au régulateur 5 V que comporte le MK 14.

Le régulateur chauffe cependant d'autant plus que la tension est élevée, et j'ai préféré lui adjoindre un petit radiateur.

Il est important d'alimenter avec une tension suffisante, supérieure à 7 volts.

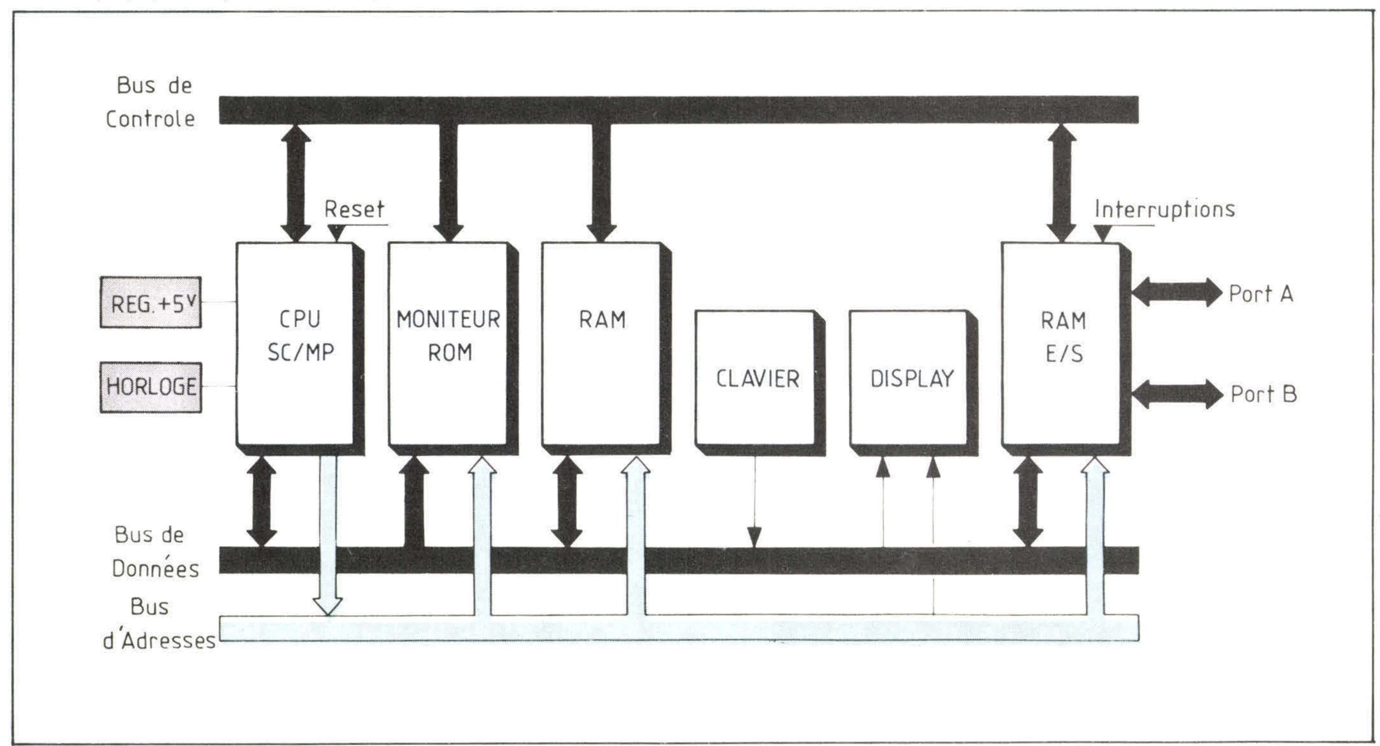
Et si, à la mise sous tension, vous n'obtenez pas les six tirets qui doivent normalement s'afficher, contrôlez la tension. Comme moi, vous aurez peut-être laissé le réglage de votre alimentation sur 5 V !

Le moniteur livré avec le système de base utilise les quatre ordres MEM, TERM, ABORT et GO, chacun correspondant à une touche du clavier.

L'utilisation du moniteur est rapidement maîtrisée grâce aux explications du manuel. Cependant, l'usage du moniteur se révèle assez fastidieux à la longue puisqu'il faut appuyer sur TERM après avoir appelé l'adresse suivante grâce à MEM.

Fort heureusement, un moniteur amélioré est disponible en option. Il devient intéressant pour un usage fréquent du MK 14 et de ses extensions.

Schéma synoptique du système. Le microprocesseur utilisé est un SC/MP de National Semiconducteur.



Interface cassette

L'interface cassette est livré en option à un prix intéressant. L'assemblage s'effectue en une heure sans problème particulier. Les liaisons avec le magnéto-cassette et le MK 14 s'effectuent par de simples fils. Il est préférable d'utiliser un connecteur à 32 broches qui servira à toutes les connexions ultérieures.

Le logiciel de lecture-écriture doit être entré préalablement en mémoire.

Mais il est déjà dans la PROM du super-moniteur, ce qui simplifie considérablement l'utilisation de l'interface.

Quel soulagement que de pouvoir se servir d'une cassette ! Les précieux programmes sont chargés à la vitesse de 100 bauds, ce qui est amplement suffisant pour ce type d'appareil.

Notons aussi que l'on peut limiter l'écriture cassette à la longueur exacte du programme si celui-ci n'excède pas 256 octets, et que l'utilisateur doit contrôler la mise en marche et l'arrêt du magnéto-cassette. Deux LED rendent la tâche aisée.

RAM d'entrée-sortie

Les possibilités d'interface du MK 14 ont été étendues grâce à l'utilisation d'une RAM I/O, en option.

Celle-ci est constituée de deux parties :

— une partie mémoire de 128 mots de 8 bits, qui s'utilisent exactement comme une zone normale de mémoire. Ils sont adjoints aux registres des entrées/sorties.

— Une partie entrée/sortie de deux ports de 8 bits.

Ce boîtier fait partie de la lignée des PIA, UIA, PPI, etc., qui ont pour tâche d'extraire les informations qui circulent sur le bus de données du système, pour les transférer vers l'extérieur.

On peut, grâce à la RAM I/O, gérer un périphérique soit par boucle d'attente grâce à la procédure du « Handshaking », soit par interruption par l'entrée « Interrupt ».

Ces deux modes permettent de brancher sur le MK 14 des équipements fonctionnant sur des entrées/sorties parallèles : imprimantes (procédure par « Hand-shaking »), convertisseurs analogiques-digitaux (procédure par interruption), etc.

Le manuel

Bien peu de constructeurs ont fait autant d'efforts pour aider les utilisateurs dans la compréhension de la micro-informatique.

Quels sont les chapitres les plus intéressants ?

Une introduction aux microprocesseurs et au langage hexadécimal est donnée à l'usage des débutants.

Des considérations sur la programmation suivent ensuite, elles rappellent la méthodologie à utiliser et donnent le jeu d'instruction du SC/MP.

Un chapitre sur l'utilisation de la RAM I/O précède enfin la seconde partie du manuel, consacrée aux programmes.

Quelle est la valeur de ce manuel ? Il guidera le débutant dans ses premiers pas. Par contre, il n'évitera pas un effort personnel qui reste nécessaire pour tout lecteur désirant acquérir la maîtrise de la programmation.

La bibliothèque de programmes

Le manuel contient plus de 20 programmes sur des sujets aussi variés que les jeux, les applications électroniques, la musique, les calculs mathématiques, etc.

Pour ma part, j'ai beaucoup apprécié la sélection de jeux, l'horloge digitale, et le programme de boîte à musique.

Sont-ils faciles à utiliser ? Un peu de réflexion est nécessaire au début, car les explications préalables, bien que suffisantes, sont brèves. Les programmes mathématiques en particulier sont traités de manière à être appelés par un programme principal qui serait développé par l'utilisateur.

Les programmes de jeux, par contre, sont sans mystère, et comprennent Mastermind et un programme d'alunissage. Ces deux programmes, étant donné leur importance, demandent des RAM supplémentaires.

Conclusion

Le MK 14 est indéniablement un bon kit d'initiation aux microprocesseurs, et un kit d'évaluation peu onéreux.

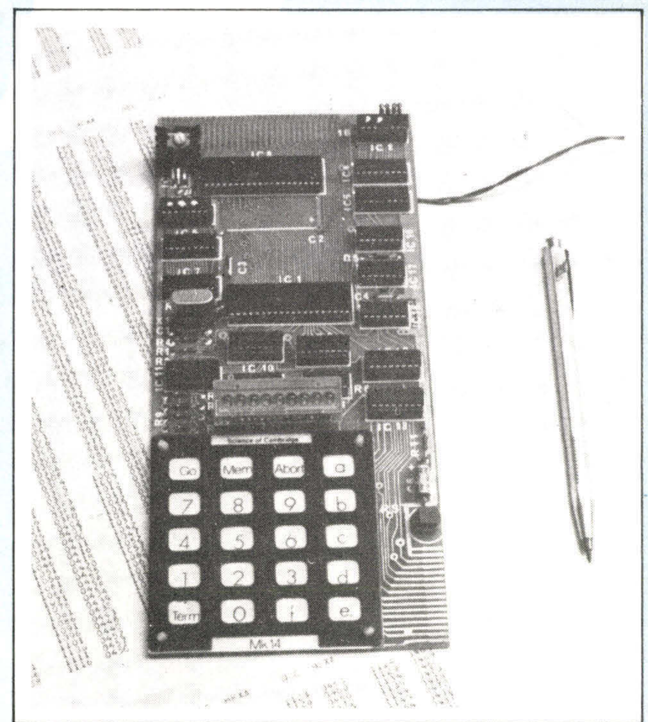
Il offre l'avantage considérable de présenter la micro-informatique comme un sujet riche et attrayant, et laisse entrevoir des domaines d'applications quasiment sans limites.

Les extensions, toutes aussi avantageuses que le kit de base, permettent enfin de poursuivre l'initiation bien au-delà de ce que ces systèmes proposent habituellement, et pour un prix très raisonnable.■

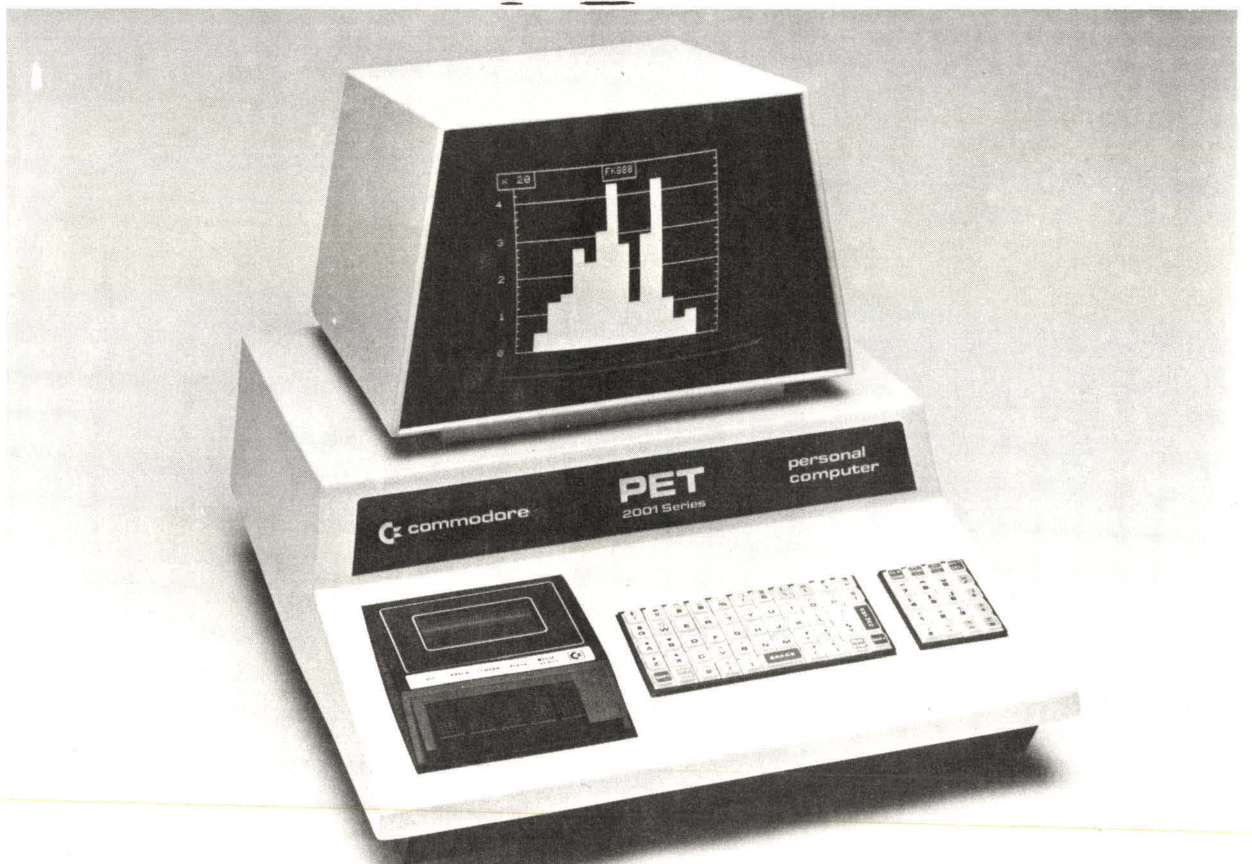
Prix : kit de base 795 F, TTC
Interface cassette 120 F, TTC
Extension RAM 198 F, TTC

** Le MK 14 est distribué en France par JCS Composants, 35, rue de la Croix-Nivert, 75015 Paris.*

La carte imprimée supporte l'ensemble des composants, le clavier et les afficheurs à diodes électroluminescentes.







Graphika

Un seul coffret
intégrant
l'écran, le clavier, le magnétophone.
le P.E.T de commodore

Complet, compact, le **P.E.T.** est particulièrement adapté à **l'enseignement**, à **l'industrie** et aux **laboratoires d'instrumentation** (bus IEEE 488). Basic puissant et rapide pour le **calcul.** Son prix le rend accessible aux utilisateurs individuels.

- Ecran incorporé à affichage très fin.
- Lecteur-enregistreur de cassettes standard incorporé.
- Clavier 73 touches avec symboles graphiques.
- Basic étendu résident avec grandes facilités d'édition.
- Interface IEEE 488.
- Connecteur d'accès à un port de 8 lignes d'entrée/sortie bidirectionnelles compatibles TTL, programmables.
- Connecteur d'accès à tous les bus du microprocesseur.

Pour 6 450 f (HT) le système complet

avec 16 K octets de ROM 9 K octets de RAM dont 7 K disponibles pour l'utilisateur

MANUEL D'UTILISATION EN FRANÇAIS

Coupon réponse à retourner à :

PROCEP 97, RUE DE L'ABBE GROULT 75015 PARIS TEL : 532.40.60.

NOM PRENOM ....

ETS ....

ADRESSE ....

....

....

TEL. ....

ms

ATTENTION NOUVELLE ADRESSE

PROCEP

97, RUE DE L'ABBE GROULT
75015 PARIS
TELEPHONE : 532.40.60.

Le MTS de I.C.S.

Kit d'initiation



Le système MTS de la société ICS

Le système MTS * se distingue d'autres micro-ordinateurs similaires par la vocation qui lui est donnée : l'aide à la formation permanente.

On notera que commercialement, le MTS n'est mis en vente qu'en tant que support de différents cours pour apprendre la micro-informatique. ICS l'emploie d'autre part, à un grand nombre d'exemplaires, comme support pédagogique lors de Séminaires donnés en public ou en entreprise.

Aspect général

Le système se présente comme une carte électronique « nue », montée sur pieds, sur laquelle sont rapportés les dispositifs de dialogue ; à savoir, un clavier hexadécimal entouré de touches de fonction (24 touches au total), et huit afficheurs à 7 segments + point.

Le montage est celui du fabricant NEC (la plupart des composants viennent de celui-ci), et on y distingue aisément quatre sous-ensembles, bien groupés sur la carte :

— l'unité centrale proprement dite, qui groupe autour d'un microprocesseur 8080 les classiques circuits d'horloge et de commande de bus ;

— des mémoires, soient quatre boîtiers EAROM (mémoires mortes qui contiennent le « moniteur » pédagogique), et huit boîtiers CMOS (mémoires vives pour l'élaboration des programmes et leurs données) ; la capacité est de 1 K octets pour chaque type de mémoire ;

— un ensemble de circuits qui réalise la fonction d'affichage, selon un procédé assez original sur lequel nous reviendrons ;

— un circuit d'entrée/sortie parallèle, dont quelques lignes sont affectées au clavier.

Une petite région reste « libre » (afin de permettre des montages particuliers). Elle est simplement pourvue de trous métallisés au pas standard. Enfin, un connecteur à 100 points renvoie les bus du système, ainsi que les lignes d'entrée/sortie utilisables.

Le système nécessite deux sources d'alimentation, aux tensions de + 5 et de + 12 volts.

Architecture du processeur

Le montage du célèbre microprocesseur 8080 est somme toute très classique. Il est pourvu de ses circuits auxiliaires traditionnels : le 8224, qui fabrique les horloges à partir d'un quartz, et le 8228, qui sépare les 8 lignes de données (data bus) et élabore les principaux signaux de commande (control bus).

Cependant, on note la présence d'un dispositif spécial, qui est conçu pour l'exécution des programmes d'application en pas-à-pas, sous contrôle du programme moniteur.

---

* *Le MTS (Microcomputer Training System) est diffusé en France par ICS (Integrated Computer Systems) 90, av. Albert-1er, 92500 Rueil-Malmaison. Tél. : 749.40.37.*

Ce dispositif fabrique une interruption après chaque instruction, ce qui permet au moniteur de reprendre en mains la « situation » entre deux instructions.

Dans le moniteur, cette possibilité est fortement exploitée ; un commutateur (AUTO/STEP) permet de désactiver le dispositif en question, pour avoir une exécution en « temps réel ».

Les mémoires

Les mémoires mortes du MTS sont d'une technologie encore peu commune, ce sont des EAROM 256 x 8 bits, qui sont effaçables et reprogrammables électriquement.

Par ailleurs, la mémoire de travail est elle aussi originale. Elle est constituée avec des boîtiers 5101 (256 x 4 bits, entrées et sorties séparées) de technologie CMOS. Cette technologie a entre autres caractéristiques de permettre le maintien de l'information avec une très faible consommation de courant.

Cette possibilité est exploitée sur le MTS : si l'on raccorde au système deux piles de 1,5 volts montées en série, et que l'on actionne un commutateur (PROTECT), le contenu de la mémoire peut être sauvegardé jusqu'à plusieurs semaines, alors que le MTS est hors tension.

L'affichage

Un autre attrait certain du système MTS est son affichage sur huit blocs LED à 7 segments + point. Tout d'abord, parce qu'il donne un certain « confort » quand on dialogue avec le moniteur, car chaque information visualisée peut être complètement identifiée ; ceci tient d'une part au nombre des afficheurs (huit), et d'autre part, au fait que tous les « graphismes » possibles sur sept segments sont accessibles.

Le montage des afficheurs est intéressant à un autre titre. Il permet, en effet, d'illustrer le concept classique de l' « accès-direct mémoire » (généralement abrégé en DMA). Sur le MTS, les blocs d'affichage sont « rafraîchis » automatiquement à partir de huit positions réservées dans la mémoire vive. Chaque bit d'une position de mémoire détermine l'allumage ou l'extinction de l'un des sept segments, ou du point. Ainsi, les programmes peuvent afficher, par simple écriture en mémoire, des « motifs » arbitraires comme le signe « = » ou la lettre « H ».

A notre connaissance, cette disposition est sans autre exemple, dans la catégorie de petits systèmes considérée.

Le clavier

Le clavier est géré suivant un procédé de balayage très classique, selon une matrice de trois colonnes x huit lignes, par l'intermédiaire d'un circuit d'interface parallèle « programmable » du type 8255. Un total de douze lignes d'entrée/sortie étant réservé pour ce faire, le circuit en question a encore douze autres points disponibles.

On relève encore quelques détails pratiques qui tendent au confort de l'utilisateur. Les touches sont de grande taille et sont « franches ». Leur marquage est très lisible, et elles sont teintées de manière différente selon qu'elles sont affectées à des « fonctions » ou à des chiffres hexadécimaux.

Utilisation

Dès la mise sous tension, ou après pression de la touche de restauration (RST), le moniteur affiche « à gauche » l'adresse hexadécimale 8200, qui est le point de départ de la mémoire vive pour les programmes. Ceci correspond au principe qui a été adopté dans le moniteur du MTS, de faire voir « par défaut » le compteur ordinal (ou : PC), autrement dit, l'adresse en mémoire de la prochaine instruction.

Une touche (ADDR) permet de modifier cette adresse ; une autre touche (MEM) permet d'activer un mode de visualisation/écriture en mémoire, par frappe hexadécimale et validation (NEXT), qui fait progresser l'adresse.

La visualisation et l'écriture dans les registres s'obtiennent par une autre touche de fonction (REG). Grâce aux huit afficheurs, le moniteur maintiendra simultanément le compteur ordinal, l'identification du registre concerné, et sa valeur. Par exemple :

8 2 0 0 H = F 3

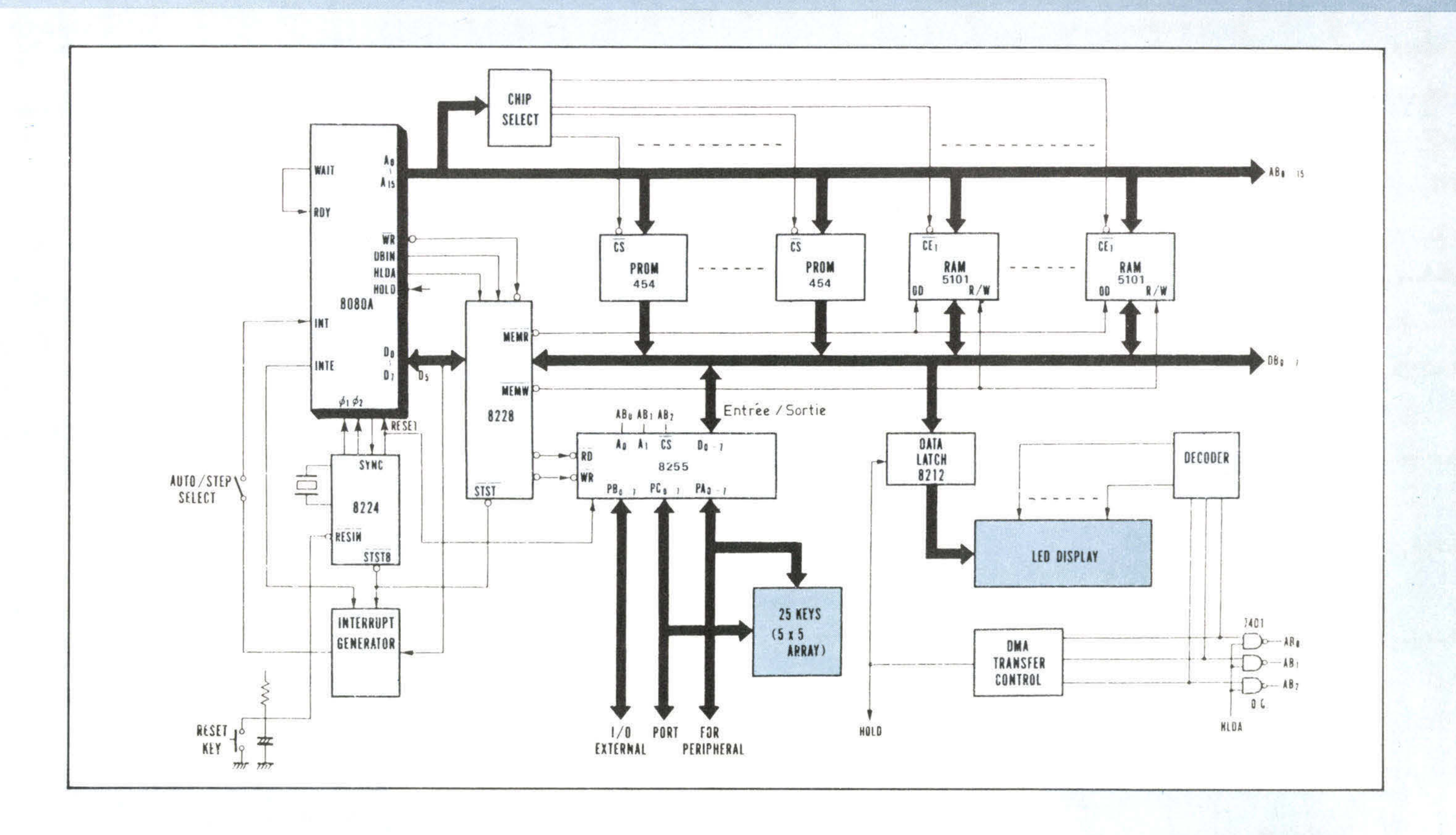
8200 : l'adresse pointée par le compteur ordinal
H : le registre sélectionné
F3 : contenu du registre H

Une caractéristique agréable du moniteur est qu'il garde cette disposition si l'on exécute un programme en pas-à-pas (STEP) ; on peut ainsi, à titre d'exemple, suivre l'évolution d'un registre en progressant dans le programme, sans aucune manœuvre supplémentaire. D'autres systèmes similaires imposent pour ce faire plusieurs frappes pour chaque instruction, ce qui est pénible à la longue. Nous voyons là un autre exemple du souci « éducatif » des concepteurs.

Nous ne passerons pas en revue toutes les possibilités du moniteur, nous contentant d'en résumer l'essentiel dans le tableau ci-après.

TOUCHE FONCTION

RST	Restauration du système (reset)
ADDR	Permet le plus généralement d'entrer une adresse. En combinaison avec les touches hexa et la touche MEM, permet de voir les paires de registres (B.C.) (D.E.) (H.L), ainsi que la position de mémoire pointée via ces paires de registres. La même possibilité existe pour le pointeur de pile ((SP) et le sommet de la pile.
MEM	Permet de valider une adresse et d'accéder à la position de mémoire correspondante.
CLR	Annule une entrée au clavier. Annule également un point d'arrêt (break point) voir BRK.
REG	Permet la visualisation et la modification du contenu des registres A, B, C, D, E, H, L et F (flags), ainsi que du pseudo-registre M (mémoire pointée via H, L).
STEP	Exécution d'un pas de programme.
RUN	Exécution « normale »
BRK	Point d'arrêt, voir suite de l'article à ce sujet.



Les points d'arrêt

Le procédé du « point d'arrêt » (en anglais : break point) est particulièrement remarquable dans le système MTS. En effet, sur la plupart des micro-ordinateurs similaires, l'arrêt se prépare par la substitution, à l'instruction normale, d'une instruction d'appel au moniteur, ceci permettant de constater le passage du programme en un point choisi. On note, en outre, que cet arrêt est généralement effectif dès le premier passage dans la séquence «piégée ».

Ceci a deux inconvénients majeurs.

D'abord, si l'arrêt ne « fonctionne » pas (c'est-à-dire que le programme ne passe « jamais » dans la séquence piégée), le programme originel n'est pas restauré, et demeure « mutilé ». Ensuite, lorsque le point d'arrêt est disposé dans une « boucle » de programme, l'arrêt sera effectif dès le premier passage, et l'observation de l'état des lieux n'apportera que peu (ou pas) d'information sur le fonctionnement répété de la boucle.

Rien de semblable sur le MTS qui surveille chaque pas de programme sur l'interruption, et n'a donc pas besoin d'altérer ledit programme pour constater le passage en tel ou tel point. En outre, le moniteur accepte qu'on lui demande de ne stopper l'exécution qu'après un certain nombre de passages au point d'arrêt, et, en conséquence, permet une bien meilleure vision des boucles de programme.

Notre avis

Si le choix du microprocesseur 8080 est discutable au plan strictement pédagogique, il donne certainement au MTS une large audience, compte tenu de la très grande diffusion de ce produit, qui se traduit par sa reprise en « n-ième » source par une quantité de fabricants.

La disposition de la mémoire vive aurait pu être plus attrayante : en effet, les 1024 positions ne sont que partiellement accessibles à cause des zones « réservées » (espace de travail du moniteur, zone d'affichage), et, d'autre part, le moniteur donne par défaut le contrôle au « beau milieu » de cet espace (adresse 8200).

Pour terminer avec les points négatifs, regrettons qu'une fois de plus un moniteur soit disposé à partir du « zéro » de l'espace-mémoire, ce qui ne donne à l'utilisateur qu'une vue imparfaite des choses...

A côté de ces inconvénients somme toute très mineurs, surtout si l'on songe à l'emploi éducatif du système, nous noterons qu'il s'agit d'un montage correct, bien réalisé et robuste (cela compte !) qui est, répétons-le, très confortable pour ceux qui le manipulent.

Le programme moniteur représente un assez joli tour de force, ce que l'on observe en énumérant, en qualité comme en quantité, les fonctions accessibles (nous n'avons pas parlé de ses sous-programmes d'entrée/sortie, intéressants à bien des égards), alors que ce même moniteur n'occupe que 1 K positions.

Le MTS nous apparaît comme l'un des meilleurs parmi les systèmes d'apprentissage qui sont proposés, au moment où ces lignes sont écrites. Il s'accompagne d'un gros livre pour apprendre seul l'essentiel sur le micro-ordinateur et sa programmation, qui donne par ailleurs des idées d'application intéressantes.

En revanche, il se place moins bien que d'autres, si l'utilisateur a pour vocation de développer autour d'un « noyau » des applications matérielles ; il est clair que ce n'est pas l'objet du MTS.

La même société ICS a développé une petite sœur du MTS, à savoir, une carte d'électronique baptisée ITS (Interface Training System) qui donnera matière à l'enseignement des techniques d'interfaçage les plus usuelles (coupleurs matériels, logiciels-types associés). Nous reparlerons de cette extension du MTS, très complète, et qui est unique en son genre à notre connaissance. ■

Jean-Michel COUR

Micro Electronique - Micro Informatique

INFORMATIQUE
D.J. DAVID

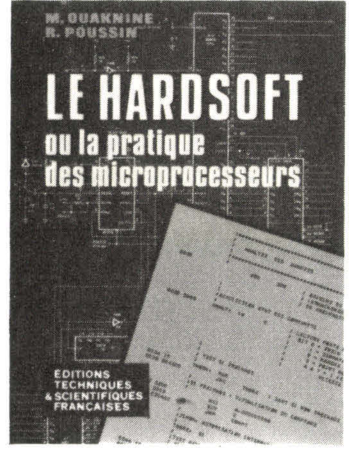
Cours d'initiation à l'informatique (ENS). Langages de programmation : Fortran, APL. Fonctionnement interne des ordinateurs. L'esprit informatique, modèles schématiques des applications, cartes-contrôle : IBM, CDC, UNIVAC, CII et Philips, 336 pages.

NIVEAU 3 — ***PRIX : 65 F***

MICRO-INFORMATIQUE MICRO-ELECTRONIQUE DICTIONNAIRE
LILEN et MORVAN (I.C.S.)

Un millier de mots, sigles et expressions. Définitions françaises et leur traduction (français-anglais). Lexique anglais-français. 370 pages.

NIVEAU 2 — ***PRIX : 92 F***



TECHNIQUES D'INTERFACE AUX MICROPROCESSEURS
LESEA et ZAKS (SYBEX)

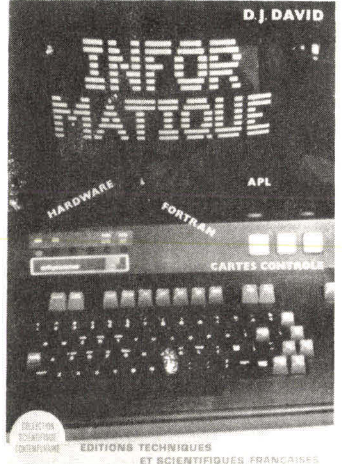
Comment connecter un système à microprocesseur aux périphériques, depuis l'unité centrale jusqu'au clavier, télétype, disque souple, écran de visualisation, et interfaces analogiques. Techniques de test. 416 pages.

NIVEAU 2 — ***PRIX : 95 F***





ÉDITIONS TECHNIQUES ET SCIENTIFIQUES FRANÇAISES
Ventes libraires. 2 à 12, rue de Bellevue
75940 Paris Cedex 19



TECHNIQUE POCHE N° 4
INITIATION A LA MICROINFORMATIQUE LE MICROPROCESSEUR
P. MELUSSON

Qu'est-ce qu'un ordinateur. Langages. Calcul binaire. Codages. Fonctions logiques. Technologie et organisation des microprocesseurs. Les mémoires. Circuits et systèmes d'interface. La programmation. 136 pages.

NIVEAU 2 — ***PRIX : 27 F***

LE HARDSOFT ou la PRATIQUE des MICROPROCESSEURS
M. OUAKNINE et R. POUSSIN

Principes généraux. Fonctionnement et jeu d'instruction d'un système construit autour d'un microprocesseur 8080A. Trois applications réelles avec schémas et programmes. Fonctionnement des dernières nouveautés 8048-Z80 - 8086. 254 pages.

NIVEAU 3 — ***PRIX : 65 F***

LEXIQUE MICROPROCESSEURS
(SYBEX)

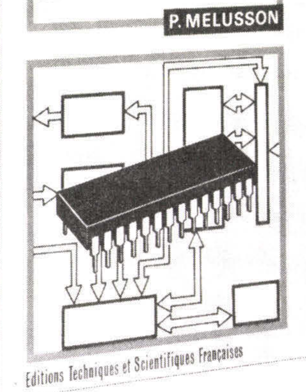
Dictionnaire anglais-français. 1 000 termes et abréviations. Définitions des composants par numéros, des signaux pour les bus S 100, RS 232C, IEEE 488. Adresses des fabricants et distributeurs. Table de conversion. Format Poche. 120 pages.

NIVEAU 2 — ***PRIX : 20 F***

INTRODUCTION AUX MICROORDINATEURS INDIVIDUELS ET PROFESSIONNELS
R. ZAKS (SYBEX)

Ce livre vous permettra d'évaluer si vous devez utiliser l'un des nouveaux microordinateurs.
Comment choisir son système.
Définitions, pièges à éviter, programmation. Quel Basic ?
— Applications professionnelles et commerciales
— Choix des périphériques.

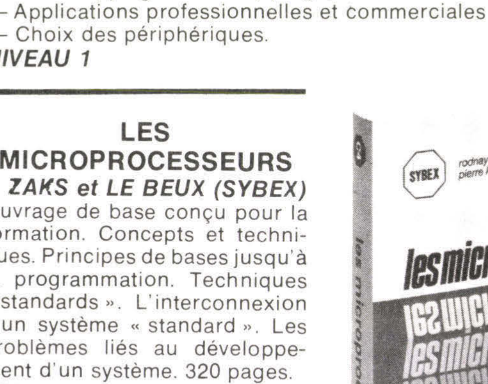
NIVEAU 1 — ***PRIX : 52 F***



LOGIQUE INFORMATIQUE
M. FERRETTI

— Qu'est-ce qu'un ordinateur.
— Cours et exercices sur la théorie des ensembles. Lois de composition. Relations binaires. Multiplication, puissance des nombres relatifs.
— L'Algèbre de Boole. Logique de commutation. Représentation et minimalisation des fonctions booléennes. 160 pages.

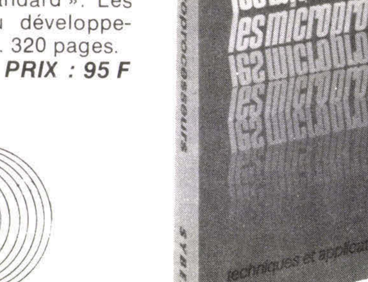
NIVEAU 3 — ***PRIX : 25 F***



LES MICROPROCESSEURS
ZAKS et LE BEUX (SYBEX)

Ouvrage de base conçu pour la formation. Concepts et techniques. Principes de bases jusqu'à la programmation. Techniques « standards ». L'interconnexion d'un système « standard ». Les problèmes liés au développement d'un système. 320 pages.

NIVEAU 2 — ***PRIX : 95 F***



ELEMENTS ESSENTIELS DE L'ELECTRONIQUE ET DES CALCULS DIGITAUX
D. ULRICH

Logique électronique. Logique informatique. Calculateurs à circuits logiques. Réalisation des calculateurs. Le transistor en commutation. Multivibrateurs. Montages logiques de base. Fonctions logiques. Algèbre de Boole. Calculs binaires. 304 pages.

NIVEAU INGENIEUR — ***PRIX : 86 F***



En vente chez votre
LIBRAIRE HABITUEL
ou à la LIBRAIRIE PARISIENNE de la RADIO
43, rue de Dunkerque, 75480 Paris Cedex 10

AUCUN ENVOI contre remboursement. Port : Jusqu'à 25 F : taxe fixe 3,50 F. De 25 F à 100 F : 15 % de la commande (+ 3,50 F Rdé). Au-dessus de 100 F : taxe fixe 18,50 F.

NIVEAU 1 : Initiation
NIVEAU 4 : Ingénieur
Tarif : Décembre 1978

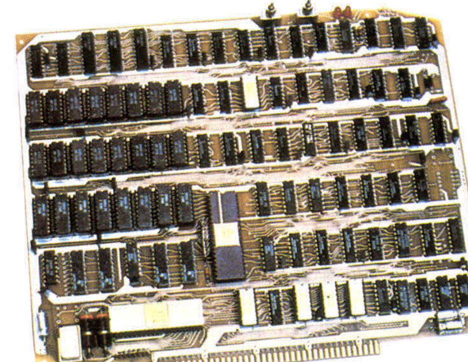
Pleins feux sur SPARK 16®

Carte Micro-Ordinateur 16 bits

Outil d'évaluation, de formation et de mise au point sur le CPU 16 bits MICROFLAME 9440. La carte SPARK 16® est plus qu'une étincelle.

Assemblée et testée sur un circuit imprimé 20,3 x 25,4 cm, elle comporte :
☐ CPU 9440* avec horloge 8 à 12 MHZ ☐ 4 K mots (8 K octets) de mémoire RAM 93481 ☐ 2 K mots (4 K octets) de PROM 93453 (moniteur) ☐ Interface avec périphérique série (14 vitesses) ☐ Contrôleur d'E/S et DMA ☐ Interface avec Console manuelle Thermostat® ☐ Expansion par bus S 100 ☐ Contrôles hardware.





Un logiciel FIRE 1

Constitué de :
☐ FIRE – BUG ☐ FIRE – LOAD : auto-chargeur
☐ FIRE – DIAG : tests CPU – tests mémoire – tests temps d'exécution – tests E/S.

MICROFLAME™
MICROPROCESSOR

La famille FIRE®, c'est aussi :

☐ 9441 Contrôleur mémoire - 32 K mots ou 64 K octets, rafraîchissement, DMA.
☐ 9442 Contrôleur entrée-sortie, 64 canaux de 3 entrées – sorties 16 bits.
☐ 93481 RAM dynamique 4 K x 1, temps d'accès 100 ns.
☐ 93483 RAM dynamique 16 K x 1
☐ 9445 CPU. MICROFLAME II (2e sem. 79) quatre fois plus puissant que le 9440

Un support logiciel FIRE

☐ FIRE 1 ☐ FIRE – EDIT
☐ FIRE – BUG ☐ FIRE – BASIC ☐ FIRE – REX
☐ FIRE – DOS.

En 1979 :
☐ FIRE – MACRO ☐ FIRE – FORTRAN ☐ FIRE – COBOL
☐ FIRE – PASCAL.



Un moniteur d'exploitation FIRE – BUG:

☐ 1,5 K mots (3 K octets) instructions
☐ moniteur E/S, chargement, dump
☐ assemblage par translation de code
☐ DEBUG.

Un BABY BASIC en PROM sur option

*Le 9440 exécute le jeu d'instructions de la ligne NOVA de DATA GENERAL.

FAIRCHILD

CAMERA & INSTRUMENT (FRANCE) S.A.

Pour toute information

Tél. : 584.55.66

Toute l'équipe de "**MICRO-SYSTEMES**" vous remercie de commencer la nouvelle année avec elle et vous adresse ses meilleurs vœux.

MICRO SYSTEMES

Comment souscrire un abonnement ?

— **par correspondance,** en utilisant le bulletin d'abonnement ci-dessous (ou une photocopie), à retourner à : MICRO-SYSTEMES, 2 à 12, rue de Bellevue, 75940 Paris Cedex 19. Tél. : 200.33.05.

— **chez votre marchand de journaux habituel,** en lui remettant le bulletin d'abonnement ci-dessous dûment rempli.

A découper suivant le pointillé

Bulletin d'abonnement

Nos tarifs :

	FRANCE	ETRANGER
MICRO SYSTEMES (6 numéros) bimestriel - 1 an	(1) ☐ 45,00 F	70,00 F

Informations :

— **pour les changements d'adresse :** joindre la dernière étiquette d'envoi, ou à défaut, l'ancienne adresse accompagnée de la somme de 2,40 F en timbres-poste, et des références complètes de la nouvelle adresse.

— **pour tous renseignements** ou réclamations, concernant votre abonnement, joindre la dernière étiquette d'envoi.

(1)

☐ Je m'abonne pour la première fois à partir du numéro paraissant au mois de :

☐ Je renouvelle mon abonnement : et je joins ma dernière étiquette d'envoi.

Je joins à ce bulletin la somme de : . . . . F par :

— chèque postal ☐ sans numéro de compte
— chèque bancaire ☐
— par mandat-lettre ☐

A l'ordre de **MICRO SYSTEMES**

(1) ☐ Mettre une croix dans les cases ci-dessus correspondantes.

Ecrire en capitales, n'inscrire qu'une lettre par case. Laisser une case entre 2 mots. Merci

Nom, Prénom (Attention : prière d'indiquer en premier lieu le nom suivi du prénom)

Complément d'adresse (Résidence, Chez M., Bâtiment, Escalier, etc.)

N° et Rue ou Lieu-Dit

Code Postal *Ville*

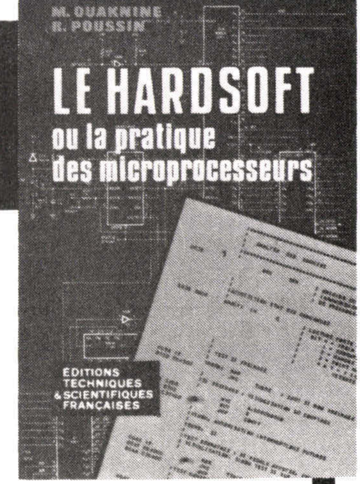
Dept	Cne	Quartier

13 *19*

Ne rien inscrire dans ces cases



EDITIONS TECHNIQUES & SCIENTIFIQUES FRANÇAISES



LE HARDSOFT

« La pratique des microprocesseurs » est un ouvrage d'initiation et de formation particulièrement destiné aux électroniciens et informaticiens non spécialistes.

PRINCIPAUX CHAPITRES

Principes généraux. Fonctionnement et jeux d'instruction d'un système construit autour d'un microproceeur 8080 A. Techniques de programmation par de nombreux exemples. 3 applications réelles avec schémas et programmes. Fonctionnement des dernières nouveautés : le 8048, le Z 80, la 3e génération. Le 8086, la 4e génération ?

En vente chez votre
LIBRAIRE HABITUEL ou à la LIBRAIRIE PARISIENNE DE LA RADIO
43, rue de Dunkerque 75480 PARIS Cedex 10

de OUAKNINE et POUSSIN
2e Edition augmentée

Les professionnels y trouveront avec profit des programmes à usage général et des schémas d'applications ainsi que des « astuces » utiles.

Un volume broché de 254 pages, format 15 x 21. 98 schémas et tableaux, couverture couleur.

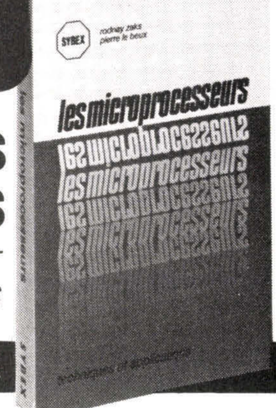
NIVEAU 3 *PRIX 28 F*
Techniciens supérieurs et dépanneurs

Commandes libraires : 2 à 12, rue de Bellevue, 75940 PARIS Cedex 19

BEST-SELLER en FRANCE et aux U.S.A.

LES MICROPROCESSEURS

TECHNIQUES ET APPLICATIONS



RODNAY ZAKS ET PIERRE LE BEUX

L'ouvrage de base sur les microprocesseurs pour toute personne ayant une formation technique ou scientifique. Il s'agit d'un livre conçu pour la formation, qui se lit facilement, malgré sa technicité. Il enseigne pas à pas tous les concepts et techniques liés aux microprocesseurs, depuis les principes de base jusqu'à la programmation. Indépendant de tout constructeur, il présente les techniques « standards » valables pour tout microprocesseur, y compris l'interconnexion d'un système « standard ». Il introduit le MPU, son fonctionnement interne, les composants d'un système (ROM, RAM, UART, PIO, autres), leur interconnexion, les applications, la programmation et les problèmes liés au développement d'un système.
Un ouvrage de 320 pages, format 16 x 24.
200 illustrations. Prix : 95 F.

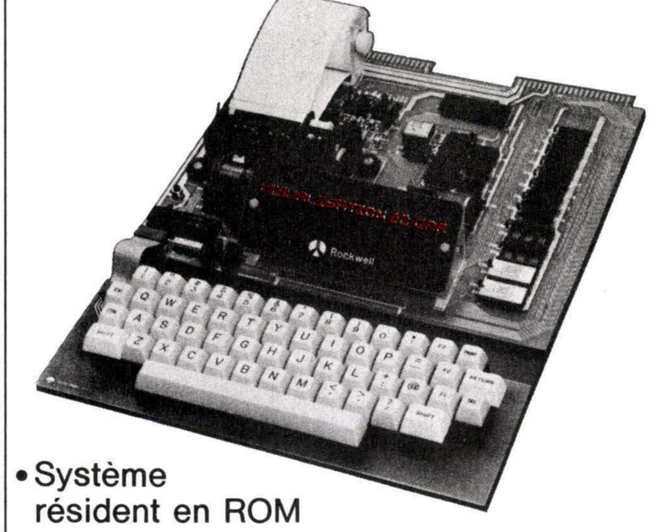
En vente chez votre libraire habituel ou à la
LIBRAIRIE PARISIENNE de la RADIO
43, rue de Dunkerque, 75010 Paris

AIM 65

un microordinateur complet

avec clavier, visualisation et imprimante alphanumériques incorporés

2990 F.h.t.*



- Système résident en ROM
- Moniteur/éditeur/pré-assembleur : 8 K
- Entrées/Sorties diversifiées : 20 lignes parallèles et série, TTY, RS232C, cassettes audio
- Timer - Intervallomètre programmable

En option : Assembleur : 4 K - BASIC : 8 K

* **Prix Janvier 1979**



EB birépub 609

REA

RADIO EQUIPEMENTS ANTARES SA

9, RUE ERNEST-COGNACQ 92301 LEVALLOIS PERRET CEDEX – TÉLÉPHONE 758.11.11 – TÉLEX 620630 F

M ....

Société ....

Adresse ....

.... Tél. ....

désire une documentation.



"DU MICROPROCESSEUR AU MICRO-ORDINATEUR" : FORMER MILLE PERSONNES EN UN JOUR ?

C'est sans nul doute plus qu'une journée exceptionnelle que viennent de vivre avec nous plus d'un millier de lecteurs de Micro-Systèmes. Alors que des centaines de visiteurs tout au long de cette journée manifestaient le plus vif intérêt pour tous les matériels présents sur les stands, mille personnes assistaient pour la première fois au plus grand séminaire technique jamais organisé jusqu'à ce jour.

Si cette manifestation était placée sous le signe de ce qu'il est déjà convenu d'appeler la micro informatique, chacun s'accorde à reconnaître que ce préfixe "micro" n'est vraiment pas de taille à qualifier l'ampleur de cet événement.

Nous tenons à remercier Monsieur Georges LECLERE, chef du service scientifique à Antenne 2 qui s'est fait un plaisir de venir projeter, en introduction du séminaire, son film tourné sur le phénomène hobbyiste aux U.S.A. et diffusé en septembre dernier sur cette même chaîne dans l'émission "question de temps".

Le point central de cette journée était bien entendu ce séminaire "Micro-processeur - Micro-ordinateur" animé de façon brillante par Rodnay ZAKS. Vous trouverez dans les pages qui suivent un large extrait de son intervention.

Avant l'ouverture du débat sur le choix d'un micro-ordinateur à usage personnel, les participants à cette manifestation ont eu le privilège de recevoir M. PELISSOLO, Directeur des Industries Electroniques et de l'Informatique, la primeur des intentions prises par les pouvoirs publics pour aider au développement en France du marché de la Micro-informatique. Nous relatons les grands moments de cette allocution dans notre encadré.

Ce samedi de la Micro-informatique devait se terminer par un débat au cours duquel se sont affrontés l'APPLE II, le PET, le PROTEUS III et le T.R.S. 80. Cette rencontre, qui ne manquât pas d'être animée sut conserver la dignité que l'on doit à une telle journée, grâce à l'arbitrage éclairé du Professeur DAVID.



Le Séminaire

Plutôt que de revenir sur le programme de ce séminaire qui vous a été largement commenté en avant-première dans notre numéro 11, nous pourrions vous présenter un résumé qui reprendrait l'ensemble des interventions de Mr. Rodnay Zaks.

En fait une telle solution ne donnerait qu'une vision beaucoup trop sommaire de cette journée, sur le plan de l'initiation à une technique enseignée jusqu'alors dans des séminaires de quelques dizaines de personnes. Il faut que tous ceux qui n'ont pas pu se joindre à nous pour cette grande première puissent avoir un aperçu de ce que peut être une telle manifestation, tant sur la qualité technique que sur la qualité pédagogique. C'est la raison pour laquelle nous dédions tout particulièrement à leur attention le compte rendu malheureusement trop court, de l'intervention de Mr. Rodnay Zaks sur l'introduction aux micro-ordinateurs, ci-après.



Rodnay Zaks, notre brillant conférencier de la journée.

Le micro-ordinateur

Quelle définition peut-on donner à l'appellation « Micro-ordinateur » ?

Un micro-ordinateur est un ordinateur dont l'unité centrale est construite autour d'un microprocesseur réalisé en un seul boîtier.

Cela veut donc dire qu'il s'agit d'un ordinateur dont la constitution repose sur des composants L S I (L S I, de l'américain Large Scale Integration). Cette technologie résulte des constants progrès apportés dans l'intégration de composants en électronique digitale. On en arrive maintenant à concevoir des circuits contenant jusqu'à 50 000 transistors ou plus sur une seule pastille de silicium de quelques millimètres de côté.

Structure de base d'un ordinateur

Il est à présent nécessaire d'envisager comment se définit la structure de base d'un ordinateur et quels en sont les éléments essentiels qui en régissent le fonctionnement.

Rappelons avant toute chose qu'un ordinateur est une machine qui permet, par l'exécution d'un programme, de traiter des données.

Ce traitement peut porter sur des calculs ou des manipulations de ces données.

Quant au programme il n'est qu'une séquence d'instructions, chaque instruction représentant un ordre destiné à traiter les données contenues dans un organe de mémoire.

Les instructions peuvent être classées en trois catégories :

Il existe des instructions arithmétiques telles que l'addition, la soustraction, la multiplication ou la division.

Il existe également des instructions logiques telles que les opérations OU et ET logiques.

A cela il faut ajouter les instructions de test et de branchement qui permettent d'orienter le programme suivant les résultats des tests.

Tout ordinateur comprend trois éléments fonctionnels de base : l'unité centrale, la mémoire centrale et les organes d'entrées/sorties.

L'unité centrale

Elle est chargée de rechercher les instructions contenues en mémoire et de les exécuter. Elle possède sa propre mémoire interne appelée registre. Ceux-ci sont conçus pour contenir les données ou les résultats d'une opération à effectuer. L'accès aux registres étant très rapide, la vitesse d'exécution des programmes s'en trouve accrue.

La mémoire centrale

Elle est destinée à stocker les programmes et les données en cours d'exécution.

Il existe différentes technologies de mémoires.

De même, on peut dire qu'il existe deux principaux types de programme chacun nécessitant une mémoire particulière qui sera décrite plus loin.

Les organes d'entrées-sorties

Il s'agit, à l'aide de ces organes, d'établir un moyen de communication entre l'ordinateur et l'utilisateur.

Pour entrer des données, l'organe privilégié est, bien entendu, le clavier.

Une imprimante ou un écran vidéo sur lequel les résultats peuvent être respectivement imprimés ou visualisés constitue un organe de sortie.

L'ensemble des composants physiques constitue le matériel **(« hardware »)**.

L'ensemble des programmes est appelé logiciel **(« software »)**.

Il n'est pas rare de rencontrer maintenant le terme américain de **« firmware »** destiné aux programmes figés à l'intérieur de mémoires mortes et qui ne peuvent être modifiés. En général « firmware » se traduit en français par **« microprogramme »**.

Le système micro-ordinateur

Nous avons vu qu'en ce qui concerne un micro-ordinateur, il s'agit d'un ordinateur dont l'unité centrale est constituée par un microprocesseur.

Dans le cas d'un micro-ordinateur de faible capacité de mémoire, l'ensemble de celui-ci, à l'exception de son alimentation, peut être réalisé sur une seule carte imprimée.

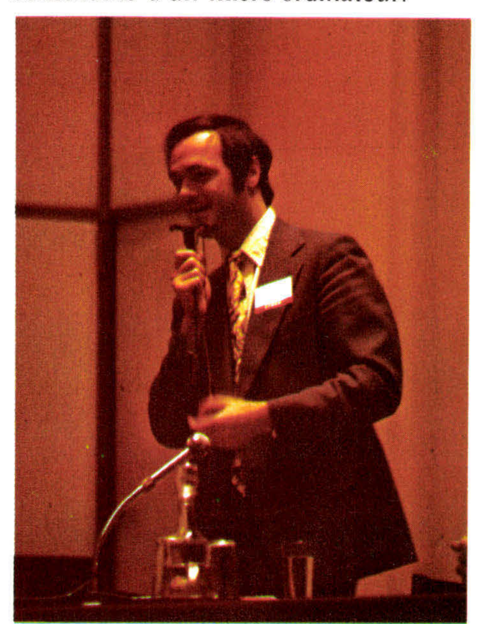
Dans la plupart des cas on arrive à disposer d'un coffret contenant l'ensemble des éléments d'extensions ultérieures.

Il ne faut pas oublier que l'adjonction d'organes externes nécessite pour chacun d'eux l'utilisation de cartes d'interface ou de « contrôleurs ».

En général, un micro-ordinateur se présente comme un boîtier compact contenant la carte unité centrale, une ou plusieurs cartes mémoire et des cartes d'entrées-sorties fournissant des interfaces périphériques standards. Il faut ajouter à cela l'alimentation.

Anatomie d'un micro-ordinateur

Nous allons maintenant examiner plus en détail chacun des trois éléments constitutifs d'un micro-ordinateur.



Un film sur les micro-ordinateurs aux Etats-Unis présenté par M. Leclere (Antenne 2).

Avant toute chose, il est important de comprendre parfaitement la fonction de chacun de ces éléments si l'on veut saisir dans son ensemble la réalisation d'un système complet. Comme dans la plupart des systèmes complexes, le micro-ordinateur doit être adapté à l'application considérée. Ceci est indispensable si l'on veut obtenir le meilleur compromis technique et économique.

L'unité centrale

L'unité centrale (« C P U » : Central Processing Unit) recherche, décode et exécute les instructions. Elle va chercher les instructions du programme dans la mémoire centrale et les exécute à l'aide de son unité arithmétique et logique.

L'unité arithmétique et logique est équipée de registres internes afin de doter cette unité centrale d'une mémoire rapide pour les données en cours de traitement.

Un programme est exécuté instruction par instruction. Normalement ces instructions sont exécutées en séquence et sont contenues dans la mémoire. Afin de mémoriser l'emplacement d'une instruction en mémoire, l'adresse de l'instruction suivante à exécuter est générée dans un registre appelé Compteur Ordinal ou Compteur de Programme (P C), interne à notre unité centrale.

Toutes ces fonctions sont réalisées sur une seule pastille : le microprocesseur.

En pratique quelques composants supplémentaires sont généralement nécessaires : ainsi l'horloge et le quartz génèrent une fréquence de référence stable.

Des composants supplémentaires sont aussi nécessaires pour amplifier les signaux : ce sont les « drivers ».

La plupart des microprocesseurs sont capables de traiter des données de 8 bits, c'est-à-dire qu'ils peuvent traiter simultanément ces 8 digits binaires.

Les bus

Le microprocesseur reçoit et transmet des informations en provenance du monde extérieur grâce à 8 lignes de données.

Ces voies de communication constituent ce que l'on appelle les **bus de données.** Un bus est simplement le groupement de lignes ayant la même fonction.

Le bus de données est bidirectionnel car l'information peut circuler dans les deux sens : vers le microprocesseur ou en provenance du microprocesseur.

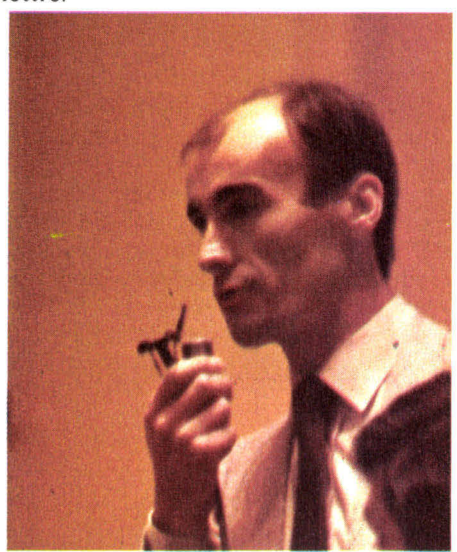
Dans le but de préciser l'origine ou la destination d'une donnée, il est nécessaire de fournir un numéro d'identification spécifié par le microprocesseur et indiquant l'adresse où se trouve la donnée.

En général, le **bus d'adresse** dispose de 16 lignes permettant ainsi de distinguer 2^{16} soit 65 536 emplacements mémoire différents.

Il est important de souligner qu'il n'y a aucune relation entre le nombre de bits présents sur le bus adresse et le nombre de bits transmis à la sortie de la mémoire et véhiculés par le bus de données.

Pour justifier ceci on peut emprunter l'analogie suivante :

Lorsque vous écrivez une lettre, vous la glissez dans une enveloppe sur laquelle vous inscrivez l'adresse qui permet de retrouver le lieu précis où doit être déposé le message contenu dans votre lettre.



J.J. Wanègue souhaite, au nom de Micro-Systèmes, bienvenue à l'assistance.

L'enveloppe constitue le bus adresse, la lettre, constitue le bus de données. Le fait que l'adresse inscrite sur l'enveloppe soit plus ou moins grande est totalement indépendant de la longueur de votre message.

Un bus de données et un bus d'adresses ne suffisent pas à rendre votre système complètement opérationnel. Compte tenu de la diversité des signaux à traiter, mais aussi de la diversité de traitements à effectuer sur ces mêmes signaux, il est nécessaire d'orchestrer tout ceci à l'aide de signaux de synchronisation. Ces signaux sont transportés par un **bus de commande.**

Ayant ainsi défini les trois types de bus connectés à un microprocesseur, nous allons pouvoir envisager les autres composants vitaux de ce système.

La mémoire

La mémoire d'un système stocke les programmes et les données sur lesquels il opère. Une mémoire est organisée en mots. Un mot est une unité logique

d'information sur laquelle le processeur travaille. Dans un microprocesseur 8 bits, le mot est donc égal à un octet (« byte » en américain). C'est pour cela que la taille des mémoires de micro-ordinateurs est généralement spécifiée en octets.

De façon idéale la mémoire d'un système doit être rapide, de très grande capacité et peu coûteuse.

Malheureusement, la vitesse et la taille ne sont pas compatibles avec un faible coût. Ceci conduit à utiliser le plus souvent deux types de mémoires : la **mémoire centrale** et la **mémoire auxiliaire** encore appelée mémoire de masse.

La première contenue dans le boîtier du micro-ordinateur est réalisée à partir de composants MOS-LSI.

La seconde est généralement constituée par un support magnétique comme les disques souples ou les cassettes magnétiques. Ce deuxième type de mémoire offre pour un faible coût des capacités de stockage supérieures à celles de la mémoire centrale mais par contre il est plus lent.

La mémoire centrale est le plus souvent constituée de deux types de mémoires : les mémoires mortes (ROM) et les mémoires vives (RAM) à accès aléatoire.

Malheureusement une mémoire vive réalisée en technologie MOS est volatile. Ce qui signifie que son contenu disparaît lorsque le circuit n'est plus sous tension. C'est pour cette raison qu'un programme permanent doit être stocké dans des mémoires mortes (« ROM » de l'américain Read Only Memory) qui ne peuvent pas être réécrites et ne sont pas volatiles.

Ces deux types de mémoires (ROM ou RAM) sont toujours nécessaires sur un micro-ordinateur. Le programme moniteur qui est l'œuvre du constructeur réside dans une ROM tandis que les programmes utilisateurs sont stockés dans des RAM à partir des disques ou cassettes où ils sont conservés en permanence.

La taille de la mémoire nécessaire dépend de l'application considérée. Une taille typique est de 2 à 8 K octets de ROM et au moins 8 K octets de RAM.

Les entrées-sorties

Le périphérique d'entrée universel est le clavier alphanumérique. C'est un clavier similaire à celui d'une machine à écrire équipé d'un certain nombre de caractères spéciaux. Il lui est associé un encodeur qui envoie directement des codes de 7 ou 8 bits vers le microprocesseur à chaque fois qu'une touche est enfoncée.

Les deux périphériques de sortie universels pour un micro-ordinateur sont l'unité de visualisation et l'imprimante.

Une unité de visualisation (écran vidéo) est pratiquement utilisée avec tous les systèmes micro-ordinateurs car elle permet de visualiser rapidement de grandes quantités d'informations.

Dans le cas où une trace permanente est nécessaire il faudra adjoindre une imprimante, cependant, une telle unité constitue un équipement coûteux.

Du fait de la complexité des périphériques, chacun d'eux exige un **contrôleur** destiné à gérer leur fonctionnement et à assurer l'interface avec le microprocesseur. Ces contrôleurs assurent le décodage des commandes et la lecture des états du périphérique. Ils permettent aussi la lecture et/ou l'écriture des données encodées dans un format associé au microprocesseur.

Avantages des micro-ordinateurs

Les avantages des micro-ordinateurs sont ceux associés à la technologie LSI.

Ce sont essentiellement :

La taille réduite

Un micro-ordinateur complet peut être réalisé dans un volume très réduit. Dans certains systèmes le micro-ordinateur peut même être incorporé directement dans le clavier du terminal.

Le coût très faible

Le coût d'un boîtier MOS est de quelques dizaines de francs. Ainsi, peut-on dire que le prix d'un micro-ordinateur dépend de la taille mémoire requise et des périphériques associés. Il est actuellement possible d'acquérir un système complet pour quelques milliers de francs.

La souplesse de programmation

Un micro-ordinateur peut réaliser des fonctions initialement obtenues à partir de moyens mécaniques, manuels ou autres.

L'avantage de la programmation est la grande souplesse que cela introduit dans l'adaptation d'un système à de nouvelles fonctions

J.-J. Wanègue

** Copyright version originale Sybex 1978 référence C1 : Introduction aux Micro-ordinateurs individuels et professionnels.*

Les intervenants de l'après-midi ; de gauche à droite : Rodnay Zaks, Isabelle Félix (chargée de mission), J.C. Pelissolo (directeur de la DIELI), J.J. Wanègue et le professeur David.



M. PELISSOLO, directeur des Industries de l'Electronique et de l'Informatique, annonce les intentions du gouvernement

Il est certain, comme le souligne M. Jean-Claude Pelissolo, que de voir ce millier de personnes sacrifier une partie de leur week-end pour venir écouter parler de microprocesseurs constitue quelque chose d'extrêmement réconfortant.

Ainsi il est souhaitable et possible à la fois que l'industrie française sache tirer profit des apports de la technologie la plus moderne au moment même où ceux-ci arrivent sur la scène internationale.

Il importe que notre industrie ne soit pas en retard d'une guerre.

Il suffit de se promener dans cette exposition qui réunit les principaux matériels existant sur le marché français pour constater la toute puissance des fabrications étrangères.

Est-ce que ce nouveau marché est à son tour en train d'échapper aux constructeurs français ?

M. Jean-Claude Pelissolo a décidé de nous convaincre du contraire, d'une part parce qu'il croit en l'existence d'une chance pour les industriels français et que d'autre part son dynamisme est de taille à pousser la chance.

Aux Etats-Unis, ce marché connaît une véritable explosion, en France il a tout juste commencé à décoller et déjà quelques produits français font leur apparition.

Quelle est la nature de ce marché ?

Il y a d'abord un marché professionnel.

Beaucoup de petites entreprises vont pouvoir enfin acquérir le matériel qu'elles ne trouvaient pas jusqu'à présent sur le marché de l'informatique, c'est-à-dire des systèmes qui soient financièrement à leur portée.

En fait, il faut adjoindre à ces petites entreprises les professions libérales (médecins, notaires, avocats...). Nous sommes en présence de ce que l'on pourrait appeler les **M.P.U.,** c'est-à-dire les **M**oyens et **P**etits **U**tilisateurs d'informatique qui constituent un vaste marché estimé à l'horizon 1985 à 150 000 micro-ordinateurs.

Il faut ajouter à ce marché celui de l'éducation et de la formation.

Et pour finir, il y a le marché personnel, celui qui, il y a encore quelques mois, était considéré comme totalement inexistant.

Quelle sera l'attitude des Pouvoirs publics face à ce marché ? Faut-il le freiner ou l'accélérer ?

A cette question, M. Jean-Claude Pelissolo offre la réponse la plus claire qui soit : « Nous ne freinerons pas ce marché, bien au contraire, nous essaierons de l'accélérer ».

Cette accélération est nécessaire car les Pouvoirs publics pensent qu'il est encore temps pour les industriels français de s'octroyer une place de choix.

« Il est souhaitable que le marché français ne soit pas dans le fourgon de queue, mais dans les wagons de tête à défaut d'être dans la locomotive... Un retard de trois à quatre ans dans l'entrée de ce marché et les industriels français perdent toute chance de recoller au peloton. »

Le Directeur des Industries Electroniques et de l'Informatique nous a ensuite fait part de trois options concrètes prises par les Pouvoirs publics.

Tout d'abord, il existe en France de nombreuses professions industrielles dont la vocation n'est pas celle de l'électronique : industrie du jouet, bâtiment, travaux publics, mécanique... Ces entreprises commencent à ressentir la nécessité d'intégrer des microprocesseurs dans leurs équipements, soit pour en diminuer le coût, soit pour créer des fonctions nouvelles et, en tout état de cause, afin d'être compétitives face à la concurrence étrangère.

Ces industriels possèdent d'excellents techniciens mais malheureusement cette technologie nouvelle constitue un handicap.

Ainsi, les séminaires sont un excellent moyen pour former des personnels nouveaux malgré leur coût encore relativement élevé au regard des possibilités financières des petites entreprises.

C'est la raison pour laquelle les Pouvoirs publics vont dans un délai de six mois mettre en œuvre une action de grande envergure permettant à ces professions de recourir à des moyens de formation accélérée, peu coûteux, et décentralisés. (La province ne sera pas oubliée.)

Deuxièmement, le ministère de l'Education va, dans un délai de cinq à six ans, introduire les micro-ordinateurs comme instruments de formation dans l'enseignement secondaire.

C'est une décision extrêmement importante qui suppose un effort budgétaire significatif, réduit grâce aux microprocesseurs, alors que les mini-ordinateurs n'avaient permis de faire à ce jour qu'une expérience limitée.

Le troisième point pour lequel M. Jean-Claude Pelissolo apportera ses soins attentifs parce qu'il s'agit d'une initiative propre à la Mission à l'Informatique dont il est le responsable ; c'est le **concours « MICRO ».**

Ce concours constitue un excellent moyen pour aider au développement de ce marché en faisant en sorte qu'à partir **d'une** activité **créative** naisse un **vaste mouvement** de micro-amateurs au même titre que nous avons eu des générations de radio-amateurs.

Le débat : PET, TRS 80, PROTEUS III, APPLE II



L'exposition, tout au long de cette journée...

Compte rendu du président de séance

La première question qui pouvait se poser est celle de savoir pourquoi on a restreint (ou étendu) le débat aux 4 machines considérées.

La réponse est un souci d'homogénéité de prix et de performances des machines à comparer. On a choisi une tranche de prix (de 5 000 à un peu plus de 10 000 F) et une gamme d'appareils.

On peut, en effet, distinguer sur le marché trois gammes de micro-ordinateurs :

— les micro-ordinateurs sur une carte, souvent en kit, qui travaillent en hexadécimal (ex : KIM, 6800 D2, etc.) ;

— les micro-ordinateurs de gestion munis d'une mémoire confortable et de tous les périphériques voulus (disques, imprimantes) dont le prix avoisine 20 000 à 30 000 F.

— Eh bien, nous avons choisi la gamme intermédiaire : micro-ordinateurs possédant une taille mémoire moyenne, munis d'un clavier complet, d'une visu, d'une mémoire de masse constituée au moins d'un magnétophone à cassettes. Cette catégorie contient les « best-sellers » du marché que sont les PET et le TRS 80, elle nous a donc semblé la plus intéressante pour une première confrontation puisqu'elle représente le juste milieu.

Un premier tour de table permet de préciser pour chaque machine la configuration hardware et software que son représentant souhaite mettre en compétition et d'indiquer son prix. Il est évident qu'il n'est pas question de donner une configuration d'appel à prix intéressant et ensuite de mettre en avant les performances d'une configuration plus chère.

Pour le **P.E.T.** présenté par MM. Kenan et Wulquin de PROCEP : le problème ne se pose pas puisqu'il n'existe qu'une seule configuration de base, et un seul niveau de Basic. Pour 6 450 F HT (7 585 F TTC) on obtient :

- l'unité centrale à base de 6502,
- le clavier 73 touches,
- l'écran (25 lignes x 40) et le magnétophone à cassettes intégrés,
- 9 K de mémoire RAM, dont 7 K sont disponibles pour l'utilisateur (1 K est utilisé par le système et 1 K sert de mémoire d'écran),
- 16 K de mémoire ROM, ainsi répartis :
  - — 8 K pour l'interpréteur BASIC étendu,
  - — 4 K pour le système d'exploitation,
  - — 2 K pour le moniteur et les routines de diagnostic,
  - — 2 K pour le générateur de caractères de la visu,
- 4 connecteurs sont disponibles et permettent d'adjoindre :
  - — des extensions-mémoire,
  - — un second magnétophone,
  - — un port de PIA (8 lignes d'entrées/sorties accessibles à l'utilisateur)
  - — un bus d'instrumentation IEEE 488.

PROTEUS : M. Perron de Proteus International met en compétition une unité centrale Proteus IIIA qui, pour 7 500 F HT comprend :

- un clavier 53 touches - UC à base du 6800,
- 16 K RAM dynamique,
- 1 K RAM de rafraîchissement d'écran.
- 8 K de ROM pour l'interpréteur BASIC.

L'unité centrale possède les interfaces télévision ordinaire et magnétophone à cassettes non compris dans le prix, puisqu'on en dispose généralement chez soi. Pour établir une comparaison valable avec les appareils confrontés qui ont ces périphériques compris dans leurs prix, il convient d'ajouter environ 1 000 F pour la télé et 300 F pour le magnétophone.

Notons que Proteus propose un moniteur spécialement étudié pour le système, Proteus Monitor au prix de 1 680 F.

L'unité centrale présente, en outre, un interface série V24 et boucle de courant.

TRS 80 : Le TRS 80 était représenté, non par le constructeur ou un importateur, mais par un utilisateur : M. Lucet, de Micrologiciels, société de software utilisatrice du TRS 80.

La configuration présentée est la 16 K BASIC niveau II qui comprend pour 6 689 F TTC :

- le clavier et l'UC basée sur un Z80,
- le moniteur (visu),
- le magnétophone à cassettes,
- 12 K de ROM pour le BASIC niveau II,
- 16 K de RAM utilisateur.

APPLE II : MM. Grigorov et Ribreau de Sonotec proposent une configuration très étendue 32 K RAM, disque souple 116 K (19 000 F HT). Pour rester dans la même gamme que les autres machines présentées ici, ils proposent une configuration plus modeste (home computer) ou, pour 9 800 F HT nous obtenons :

- le clavier et UC (basée sur le 6502),
- 16 K octets de RAM,
- 8 K ROM comprenant :
  - — un mini assembleur (2 K)
  - — un BASIC en arithmétique entière (6 K)
- l'interface magnétophone à cassettes et télévision couleur SECAM. (Le magné-

Quel micro-ordinateur choisir ?

tophone et la télé n'étant pas compris dans le prix, il faut, comme pour Proteus, ajouter environ 1 300 F pour effectuer des comparaisons valables).

• La carte unité centrale présente 5 emplacements pour fixer des cartes d'extensions ou d'interfaces particulières.

On demande ensuite à chaque représentant de citer brièvement les points forts de son matériel.

Points forts

Les points forts du **P.E.T.** sont essentiellement :
— la présentation compacte, en un seul coffret robuste, sans fils de liaison entre modules ;
— la finesse de l'unité de visualisation et la stabilité de l'affichage qui permettent de travailler longtemps sans fatigue ;
— la fiabilité et la commodité d'emploi de la cassette. Noter aussi la possibilité de connecter directement un second magnétophone identique : le software de sa gestion est déjà présent ;
— le BASIC du PET est un des plus complets et le plus rapide de cette classe de matériels (3 à 4 fois plus vite que l'IBM 5100) ;
— les interfaces présents sur la carte, notamment le bus IEEE.

Pour le **PROTEUS,** on peut noter :
— qu'il est fabriqué en France ;
— qu'il est bâti autour du 6800 et a un contrôleur de DMA ;
— qu'il gère une visu 16 lignes x 64 et a des caractères semi-graphiques analogues à ceux du P.E.T. ;
— qu'il peut gérer des unités de disques souples double densité (non compris dans le prix de la configuration envisagée ici) ;
— le BASIC occupe 8 K de ROM et a une précision de 9 chiffres significatifs.

D'autres points ont été mis en avant, mais on a pu remarquer qu'ils étaient disponibles sur les trois autres micro-ordinateurs confrontés. Ce sont :
— gestion de la cassette par modulation de fréquence pour augmenter la fiabilité ;
— utilisation d'un BASIC interprété et non compilé.

Pour le **TRS 80** les points essentiels sont :
— le prix ;
— le système est modulaire et basé sur un microprocesseur Z 80 ;
— l'interpréteur BASIC fait 12 K : il est donc très complet (par exemple : 12 fonctions de chaînes de caractères, 16 fonctions arithmétiques, calculs sur matrices ; certaines variables flottantes peuvent avoir 16 chiffres significatifs) ;
— le système étant très répandu du logiciel existe et on peut trouver des périphériques développés par des maisons indépendantes.

Pour **APPLE II,** on peut signaler particulièrement :
— les possibilités d'extension : des connecteurs pour toutes sortes de cartes supplémentaires existent dans le coffret ;
— le software :
• 2 K moniteur, mini-assembleur, simulateur de machine 16 bits
• 6 K interpréteur BASIC limité aux entiers compris entre − 32 767 et + 32 767.

Au présentateur qui considère qu'il s'agit là d'un avantage, le président de séance croit devoir faire remarquer que, en comptabilité où on doit tenir compte des centimes, cela limite à une somme de 327 francs 67 !!
— l'affichage graphique. La mémoire d'écran permet de stocker à la fois du texte et des informations graphiques haute résolution (280 x 192), en couleurs (5 couleurs par point) si la télévision utilisée est en couleur.

Nous abordons ensuite le débat proprement dit où sont posées diverses questions, auxquelles chacun des présentateurs répond tour à tour. Le dernier point fort cité de Apple conduit tout naturellement à la première de ces questions.

Quelles sont les possibilités graphiques de vos matériels ?

Apple : Cf. ci-dessus : résolution 280 x 192 en couleurs.

P.E.T. : l'écran affiche 25 lignes de 40 caractères en matrice 8 x 8, lettres majuscules et minuscules. Un jeu complémentaire de caractères semi-graphiques très bien étudié (il a, depuis, fait école) offre des possibilités très intéressantes. Tous les caractères sont affichables en contraste inversé.

Il existe certains moyens d'augmenter la résolution **apparente** de l'écran, en employant certains des caractères graphiques : on peut par exemple, faire du 50 x 80 à l'aide des caractères ▖ ▞ etc. du 200 x 40 (_ ‒ ‾ etc.) et du 50 x 320 (▏ ▕ etc) ce qui permet d'avoir une bonne approximation d'une courbe.

On peut enfin en changeant la ROM génératrice de caractères, utiliser un autre jeu adapté au tracé de courbes : on en avait un exemple à l'exposition.

Un autre avantage : on peut par des POKE, éteindre et allumer l'écran sans toucher à la mémoire de rafraîchissement, ce qui permet de créer des dessins animés saisissants.

TRS 80. L'écran affiche 16 lignes de 64 caractères. On peut former de grandes lettres. Il existe des fonctions graphiques permettant d'allumer ou d'éteindre des taches dans une matrice 48 x 128.

Pour un supplément de 100 F, on a accès aux lettres minuscules.

Proteus : Des caractères pseudo-graphiques sont disponibles. On a aussi la possibilité de changer la ROM génératrice de caractères.

Quelles sont les facilités d'édition de votre matériel ?
(Ou : j'ai une erreur dans une instruction, dois-je la retaper entièrement ?).

Proteus : pas de facilités d'édition : il faut retaper. Mais les erreurs de syntaxe sont décelées à la frappe.

P.E.T. : Toutes facilités d'édition : on a tous les mouvements de curseur sur l'écran. On peut donc se positionner sur la zone erronée et procéder à toute insertion, suppression au remplacement de caractères. Dès que l'on fait **retour chariot** la version corrigée est enregistrée.

Apple-TRS : Mouvements de curseur et facilités à peu près analogues.

Accès au langage-machine

On peut avoir besoin d'utiliser un sous-programme écrit en langage machine pour des questions de vitesse d'exécution. L'accès au langage machine suppose trois choses :
1 – possibilité d'appeler un sous-programme en langage machine à partir du BASIC ;
2 – possibilité d'entrer ce sous-programme en mémoire ;
3 – possibilité d'écrire ce sous-programme en assembleur symbolique ce qui est plus commode.

Sur le point 1, les 4 machines offrent les mêmes facilités : on dispose d'une instruction CALL, SYS ou USR permettant d'appeler un sous-programme machine dont on fournit l'adresse. Le **TRS 80** limite à 10 le nombre de tels sous-programmes utilisables simultanément.

Sur le point 2, on peut toujours entrer le sous-programme sous forme de table avec des POKE (rappelons que POKE x, y signifie « écrire x à l'adresse y »). Chacun des systèmes dispose en outre, sur cassette, d'un **moniteur** permettant l'entrée d'un sous-programme en hexadécimal, sa sauvegarde sur cassette, etc. Celui du **P.E.T.** permet la mise au point de la routine machine par exécution en pas-à-pas.

Pour le point 3, chacun des systèmes possède sur cassettes un ou plusieurs assembleurs (payants).

Comment est organisé votre service après-vente ?

Pour le **P.E.T.,** la garantie a été portée à 6 mois. La maintenance est entièrement assurée en France. Le délai moyen de réparation est de 6 jours, même en province, là où il y a un représentant.

Pour **Proteus** la garantie est de 1 an. On procède par échange standard de la carte-mère, dont chaque revendeur est tenu de stocker un exemplaire.

Pour le **TRS 80** la garantie est de 90 jours. Il y a un technicien dans chaque point de vente Tandy (actuellement 15). La réparation prend, en moyenne, 15 jours à 3 semaines. Un système de remplacement est mis à disposition des utilisateurs commerciaux pendant ce délai.

Pour l'**APPLE,** la garantie est de 6 mois. Pour les réparations, on procède par échanges standards.

Votre système possède-t-il une horloge temps réel ?

Seul le **P.E.T.** possède une horloge temps réel parmi les configurations confrontées. Pour les autres systèmes, il faut passer à un niveau supérieur. Par exemple, pour le **TRS 80,** il faut acquérir le coffret « d'interface » (coût 2 090 F TTC) qui permet des extensions de mémoire et mini-disques (on ne l'achète pas uniquement pour l'horloge !). Notons que cet achat est indispensable pour toute extension, à partir de la configuration à 6 700 F, donc si on veut dépasser 16 K, ou connecter un minidisque.

L'horloge T.R. du **P.E.T.** consiste en deux variables spéciales TI$ qui donne l'heure sous la forme HH, MM, SS et TI qui est mise à zéro au départ et incrémentée tous les 1/60e de seconde. Ceci permet des jeux ou des programmes d'entraînement avec réponses en temps limité, ou encore des applications tel que « couper le courant dans tous les bureaux après 18 heures ».

Y a-t-il un club d'utilisateurs de votre matériel ?

Proteus et **Apple** n'ont pas de club spécifique mais des clubs connus utilisent, entre autres, leur matériel.

Le **TRS 80** n'a pas encore de club en France, mais en a aux Etats-Unis.

Le club **P.E.T.** France existe d'ores et déjà. Il publie un bulletin. Il a pour but de faciliter les échanges de programmes et d'astuces d'utilisation entre les adhérents et il centralise les informations issues de clubs étrangers, notamment le club anglais.

Le président de séance pose maintenant aux participants des questions plus provocatrices résumant des critiques lues dans la presse. Il offre ainsi l'opportunité de répondre à ces critiques.

Apple :

Q – Dans un article, par ailleurs critiqué pour sa violence, un commentateur de « Byte » a qualifié le plastique de Apple de peu solide. Le commentateur est même parvenu à casser le coffret. Quelle est votre réponse ?

R – Nous n'avons jamais eu de tels problèmes. Bien sûr, et c'est d'ailleurs précisé, Apple n'est pas fait pour qu'on tape dessus avec un marteau !

TRS 80 :

Q – J'ai lu dans la presse que le magnétophone du TRS 80 est peu fiable et malcommode à utiliser. En particulier, pour rebobiner ou positionner la cassette par un fast-forward, il faut déconnecter la cassette de l'unité centrale. Cela me semble risqué d'avoir à manœuvrer la prise si souvent.

R – On peut s'en dispenser : il suffit de taper la commande CLOAD.

Q – Mais alors cela interdit la pratique suivante : on charge un programme ; pendant qu'il s'exécute, on rebobine la cassette, on la range, on installe la prochaine cassette à lire, éventuellement, on la positionne ; cela gagne du temps.

R – Le procédé est pratique mais pas indispensable.

Q – Autre critique adressée à la cassette du TRS : le compteur est très imprécis de sorte que, même si on se positionne au bon chiffre, on risque d'être plus loin que le début du programme à lire.

R – Il est prudent de se positionner un peu avant.

P.E.T. :

Q – Il n'y a pas de compteur au magnétophone du P.E.T.

R – En effet et cela nous a nous-mêmes gênés au début. Depuis, étant donné qu'on peut commander le moteur du magnétophone par programme, des procédés ont été mis au point permettant de faire un déroulement rapide de la bande pendant un temps chronométré par le P.E.T. lui-même. D'autre part, la commande VERIFY permet de se positionner avec précision juste derrière un programme. Par ailleurs, cette commande est très pratique, puisqu'elle permet de vérifier, par comparaison avec la mémoire, qu'une sauvegarde est correcte.

Un tour de table est effectué pour savoir si les autres systèmes ont une commande analogue :

TRS 80 : oui ; elle s'écrit CLOAD ?

Apple : non.

Proteus : non ; Proteus gère le magnéto de façon fiable et n'en a pas besoin.

Un échange de vues permet alors de préciser que le **P.E.T.** sauve les programmes (et les fichiers) sur cassettes avec un nom qui peut atteindre 16 caractères, que le **TRS 80** ne tient compte que des 2 premiers caractères du nom, que **Proteus** ne tient compte que d'un caractère et que **Apple** n'utilise pas de nom.

On laisse maintenant la parole dans la salle pour quelques questions.

Q – Lequel de ces systèmes pourrait remplacer un automate programmable ?

R – A notre connaissance tous.

Q – Existe-t-il des notices en français de ces matériels ?

R – Proteus : d'office en français. **Apple :** oui. **TRS 80 :** en cours de traduction. **P.E.T. :** oui.

Q – Pourquoi voit-on de telles différences dans le prix du dollar rendu en France ?

R – Trois causes possibles, sous toutes réserves :

1 – des différences de taille de maison entraînant des frais différents, etc.

2 – certains prix d'attaque sont pratiqués sur le marché américain intérieur, mais non à l'exportation.

3 – certains importateurs peuvent chercher à augmenter leurs bénéfices, pourquoi pas ?

Q – Existe-t-il, ou existera-t-il un marché de l'occasion pour ces matériels ?

R – C'est aux utilisateurs futurs de dire s'ils ont l'intention de revendre leur matériel. Il sera toujours plus facile de s'étendre dans la même marque de matériel.

De multiples questions resteraient encore à poser mais le temps presse de conclure. Le président de séance remercie les participants d'avoir répondu aux questions et il souhaite que leurs réponses aient permis d'éclairer le choix des personnes désireuses de s'équiper.

C'est en dernier ressort à l'utilisateur qu'appartient la décision finale, car c'est lui qui connaît ses applications. Pour cette raison, le président de séance refuse d'indiquer ses préférences même si elles sont notoires car elles sont spécifiques de ses applications. On finit par lui arracher l'aveu que son système préféré est le Control Data 7600 ! ■

Le Président de Session
Le Professeur DAVID

Conclusion

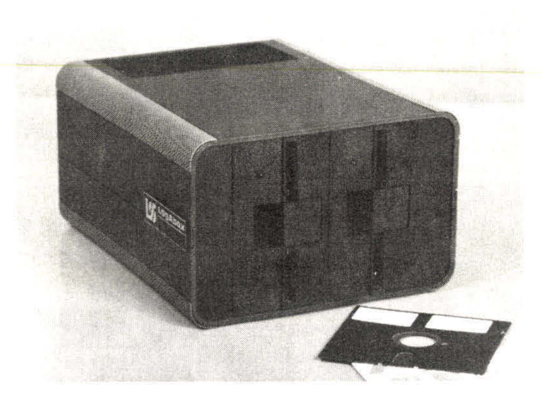
Il est désormais démontré que plus de mille personnes peuvent être formées simultanément par un seul conférencier aux principes des micro-ordinateurs et ce, malgré la grande inhomogénéité de l'auditoire.

Si vous n'avez pas participé à la journée du 25 novembre et si vous souhaitez que nous programmions à nouveau ce type de manifestation, nous serons heureux de recevoir vos suggestions à ce sujet.

Si vous êtes parmi ceux qui nous ont tenu compagnie durant cette journée, quel séminaire aimeriez-vous que nous vous présentions à nouveau ?

Ainsi, Microsystèmes remplira parfaitement sa vocation : vous permettre de maîtriser ces nouvelles techniques.

L'ordinateur personnel français.



Une technologie maîtrisée, la volonté permanente d'innover et la connaissance approfondie des besoins en informatique des entreprises et des individus ont permis à LOGABAX de mettre au point le premier ordinateur personnel français : le LX 500.

Compact, d'un prix modique eu égard à ses capacités et ses performances, facilement utilisable par des non spécialistes dans leur cadre professionnel, le LX 500 se présente dès aujourd'hui comme une famille de produits;

– LX 510 – 11.000 F H.T.* – constitue la version de base : une unité centrale à microprocesseur, 1,5 K octets de mémoire morte (ROM), 16 K octets de mémoire vive (RAM), 2 entrées/sorties aux normes V-24 du CCITT, une unité de mini-disque souple, disquette de 5 1/4 pouces, capacité 90 K octets.

– LX 515 – 14.000 F HT* – Système comprenant une deuxième unité de disque souple : capacité de la mémoire auxiliaire portée à 180 K octets.

– Extension de la mémoire vive de 16 K octets, portant la capacité totale de mémoire interne à 32 K octets – 3.000 F H.T.*.

– LX 600 – 9.600 F H.T.* - Terminal clavier imprimante, clavier ASCII, imprimante thermique à matrice 5 x 7, 80 colonnes, vitesse 30 cps.

La famille LX 500 dispose d'un logiciel complet comprenant un système d'exploitation BDOS permettant les fonctions fondamentales nécessitées par la présence d'un disque et d'un langage de programmation : le BASIC.

L'initialisation automatique dès la mise sous-tension libère l'utilisateur d'un dialogue complexe avec le système et lui permet de se consacrer exclusivement à l'application.

*Prix valables au 1/11/78.

LogAbax
informatique

Premier constructeur français de mini et péri-informatique.

Bureau de Vente, 146 Av. des Champs-Élysées - 75008 Paris. Tél. 359 61 24

Six leçons pour programmer

La plupart des notions que nous avons introduites jusqu'à présent sont, sinon universelles, du moins très générales. Un certain nombre des « unités centrales » du type microprocesseur sont construites selon les mêmes principes :

- **manipulation de l'information sous forme binaire codée ;**
- **circulation de l'information entre les mémoires, les « ports » d'entrée/sortie et l'unité centrale, grâce à des voies de données, d'adresses et de commandes ;**
- **programmation par le jeu d'une suite d'instructions enregistrées dans une mémoire, et que le processeur « appelle » et exécute successivement.**

Si nous en restions à ce niveau de description, nous pourrions croire que les microprocesseurs sont tous à peu près semblables puisque, de surcroît, ils sont fabriqués selon des technologies très voisines.

On peut commencer de les différencier par la taille du « mot » (4 bits, octet, 12 bits, 16 bits...) qu'ils traitent en une seule opération. Cette première division en catégorie, quoique essentielle, demeure fort grossière (comme de distinguer, parmi les véhicules, les « deux-roues », les voitures particulières et les « poids lourds »).



L'instructor 50 est organisé autour du microprocesseur 2650 dont nous étudions la programmation.

Pour caractériser plus finement un processeur, nous devons considérer en détail son **architecture,** c'est-à-dire ses **ressources** (moyens de stockage et de traitement de l'information), ses **opérations** (ce que l'on peut lui demander de faire avec ses ressources) et ses **modes d'adressage** (les procédés possibles pour répéter une information dans son espace de mémoire).

L'ensemble de ces caractéristiques se regroupe tout naturellement, comme nous allons le voir dans la description du **répertoire des instructions (*)** acceptées par le microprocesseur. Le mot « répertoire » est particulièrement bien venu, puisqu'il évoque irrésistiblement l'ensemble des « morceaux » que savent « exécuter » un musicien, ou mieux encore, un orchestre. Ce répertoire est la véritable « marque de fabrique » d'un processeur, en même temps que son **mode d'emploi** détaillé.

** En anglais, on dit plus prosaïquement « instruction set », soit, littéralement, « ensemble des instructions ».*

Les schémas des instructions

Dans une recette de cuisine, l'expression « faire revenir les oignons » s'analyse en : « Faire revenir » **(l'opération)** et « les oignons » **(l'opérande).** De plus, la recette fait appel **implicitement** aux moyens de réaliser cette étape de la recette : réchaud, sauteuse...

De même, la plupart des instructions d'un répertoire comme celui du microprocesseur 2650 comportent schématiquement deux parties (deux paquets de bits) :

— le **code-opération** (que nous abrégerons en **code-op.)** qui dit QUOI FAIRE ;

— les **champs d'adresse,** qui disent AVEC QUOI.

Selon la quantité d'information nécessaire pour désigner complètement les opérandes (peu s'il s'agit de registres, davantage s'il faut désigner une position de mémoire ou un élément d'entrée/sortie) les champs d'adresse seront plus ou moins grands.

Pour le 2650, l'essentiel des instructions comporte un code-opération de 6 bits, complété de 2 bits qui désignent soit un **premier opérande** (registre), soit une **variante** du code-op. (par exemple, une condition). Ce qui constitue le premier octet ; et dans certains cas, le seul (d'une instruction).

code-op						R/V	
7	6	5	4	3	2	1	0

Selon différentes techniques de désignation d'un second opérande (il en faut bien deux, pour une addition, par exemple), il faudra pour obtenir une instruction complète :

— un second octet, en particulier si ce second octet est le deuxième opérande, ou

— deux octets supplémentaires, si l'on doit **adresser** un emplacement mémoire (ce qui exige 13 ou 15 bits pour le 2650).

Ainsi, le 2650 a dans son répertoire, trois schémas de base, ou **formats** d'instruction, caractérisés par leur taille, et résumés sur la **figure 1.**

Dans le 2650, les instructions comportent un code opération de 6 bits complété de 2 bits qui désignent soit un premier opérande soit une variante du code-op.

Format	Nbre octets	Type d'opérande
[1 octet]	1	Le plus souvent, un registre
[2 octets]	2	Valeur sur un octet ou emplacement de mémoire désigné par une adresse « réduite »
[3 octets]	3	Emplacement de mémoire désigné par une adresse « complète »

Figure 1.

C'est le code-op. qui est « lu » le premier par le microprocesseur, et qui lui « annonce » si l'instruction est à 1, 2 ou 3 octets ; au besoin, le microprocesseur lira donc un ou deux octets supplémentaires en séquence, avant de procéder à l'exécution proprement dite. On doit constater, au passage, qu'un microprocesseur consacrera approximativement la moitié de son activité (et des accès à la mémoire) à la lecture et à l'analyse des instructions.

Fig. 2. – Organigramme de recopie sur des voyants d'un code entrée à l'aide de clés.

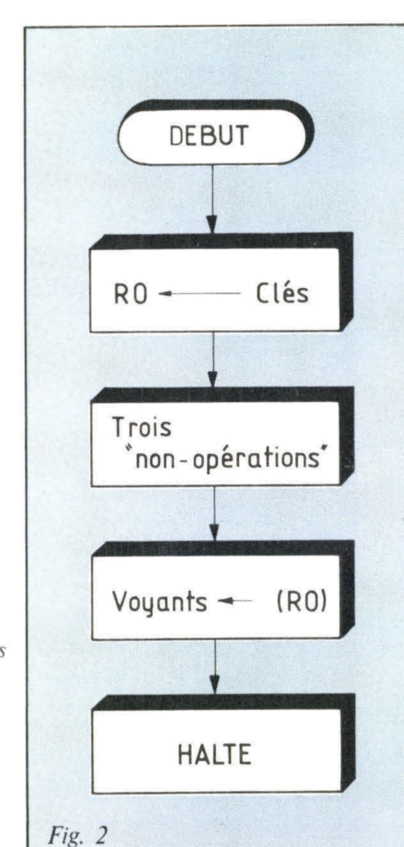


Fig. 2

** Pour toute la suite, vérifier que le commutateur d'interruption (face dessous) est dans la position KEYBOARD.*

** Mnémonique : ... « qui aide la mémoire » (Larousse) ; ce n'est pas du franglais comme certains le croient.*

** Combien de numéros de téléphone retenez-vous par cœur ? et combien de noms de personnes ?*

Un programme pour l'étude du répertoire

A notre habitude, nous recourrons à l'expérimentation plutôt qu'à la théorie, pour l'étude du répertoire de notre microprocesseur. Aussi allons-nous procéder à la « mise à feu » de l'I 50 * et y entrer un petit programme très simple ; dans ce programme, nous ménagerons des emplacements pour placer successivement des instructions nouvelles.

Ce programme s'énonce comme suit :

- Lecture des clés du pupitre dans le registre 0 ;
- « trois mesures pour rien » ;
- écriture du contenu du registre 0 sur les voyants ;
- arrêt du microprocesseur (halte).

Ou, sous forme de l'organigramme de la **figure 2.**

Pour sa traduction en « binaire », nous adopterons une présentation classique selon 6 « colonnes de texte, telle que celle donnée **tableau 1.** Nous sommes déjà (dans une certaine mesure) familiers avec les deux colonnes de gauche, qui donnent les **adresses d'implantation** du programme et la **traduction** (code-objet), ainsi qu'avec la colonne de droite, celle des **commentaires** illustrant le programme. Expliquons un peu les trois autres colonnes.

La colonne **« étiquette »** sert à **nommer** des « points de passage » (ici, le DEBUT et la FIN du programme) et/ou des emplacements de mémoire. Il s'agit là d'un artifice pour faciliter la rédaction et la lecture des programmes sous une forme symbolique et qui ne laisse pas de trace directe dans la traduction.

La colonne « opération » est, elle, directement déterminante pour la traduction, car c'est là que s'énoncent les **noms mnémoniques (*)** des codes-op. Notre esprit préfère (c'est comme ça !) retenir des noms de baptême plutôt que des chiffres, et en particulier des chiffres binaires *. Aussi chacun des codes-op., dans un répertoire d'instructions, reçoit-il un nom mnémonique... du moins pour le lecteur de langue anglaise !

Ainsi, dans notre programme :
REDD comme **RE**a**D D**ata, « lecture de donnée » (instruction qui lit nos clés).
NOP comme **N**o-**OP**eration, « non-opération » (instruction qui ne fait rien d'autre que prendre le temps d'être vue par le processeur).
WRTD comme **WR**i**T**e **D**ata, « écriture de données » (instruction qui écrit sur les voyants).
HALT qui s'explique d'elle-même !

Pour le moment, la colonne « paramètres » est vide car nos quelques instructions sont si simples qu'elles sont toutes réduites à leur seul code-op.

Cette présentation est généralement admise par toute la communauté informatique ; nous l'emploierons pour sa commodité, et nous conseillons au lecteur de l'utiliser. Le tout est d'adopter une **présentation standard** et bien ordonnée !

Opération entre registres

Avant de manipuler, vérifiez que le commutateur d'interruption de la face dessous est dans la position KEYBOARD ; disposez aussi le commutateur voulu sur NON-EXT DATA PORT. Vous pouvez

Tableau 1

Programme : LEÇON 3			**Auteur :** JM COUR	**Date :** 11/78	**Page :** 1/1
Adresse	**Traduction**	**Etiquette**	**Opération**	**Paramètres**	**Commentaires**
0 0 0	7 0	DEBUT	REDD, R0		R0 ← clés
0 0 1	C 0		NOP		
0 0 2	C 0		NOP		} 3 « non-opérations »
0 0 3	C 0		NOP		
0 0 4	F 0		WRTD, R0		Voyants ← (R0)
0 0 5	4 0	FIN	HALT		Arrêt
0 0 6					

maintenant entrer le programme en mémoire :

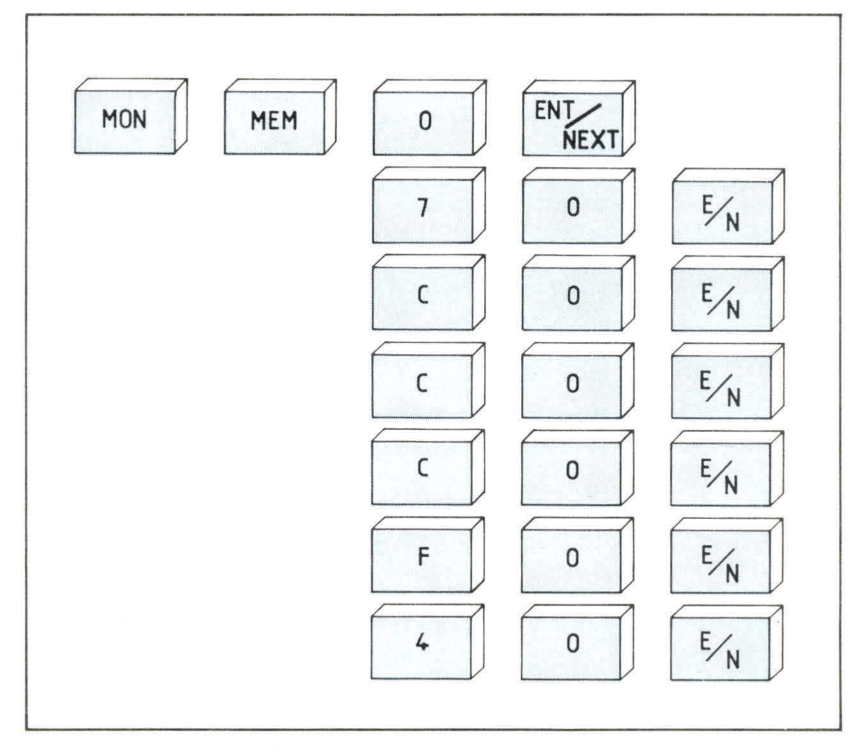


Fig. 3. – Pour entrer en mémoire le programme de recopie, il vous suffit de frapper successivement sur le clavier les touches indiquées.

Le programme s'exécute depuis l'origine par RST, et l'on constate que :

- les clés sont « recopiées » sur les voyants ;
- le témoin auxiliaire RUN s'éteint, ce qui signifie que le processeur est arrêté (HALT !).

On peut recommencer le programme autant de fois que l'on veut par RST, et vérifier que la recopie est fidèle : en effet, les instructions NOP ne font (par définition) aucun **traitement** entre la lecture et l'écriture de l'octet.

Une première façon d'intervenir, autrement dit, d'insérer un traitement entre l'entrée et la sortie, sera de combiner l'octet avec le contenu d'un des **registres généraux** du 2650.

Le microprocesseur comporte en effet une (très) petite mémoire de travail interne, agencée selon la **figure 4.**

A tout moment, le programme dispose, dans le 2650 même, de **quatre** cases d'un octet, les registres R0, R1, R2, R3 ; une « doublure » des registres 1, 2 et 3 est aussi disponible, par un mécanisme de commutation que nous laisserons temporairement de côté ; retenons seulement qu'à un instant donné quatre registres seulement sont « visibles ».

Une série d'instructions du 2650 permet de combiner le registre **Zéro** et l'un quelconque des

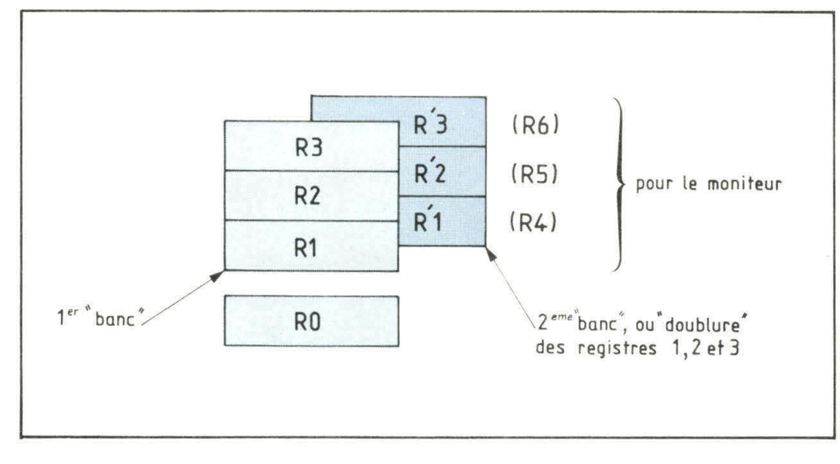


Fig. 4. – Registres généraux du 2650. Le microprocesseur dispose à tous moments, pour exécuter son programme, de 4 cases de 8 bits.

L'adressage relatif indique sur un seul octet la différence entre les adresses de l'emplacement de l'instruction et de l'opérande.

registres. Elles n'occupent qu'un seul octet et se codent comme suit :

code-op.	R
7 6 5 4 3	2 1 0

Ainsi, le code-op. d'une addition avec registre est 100000, et l'addition du contenu du registre 1 au registre 0 donne l'octet d'instruction :

|1,0,0,0,0,0|0,1|
code-op. ADDZ — registre n° 1

Soit 81 en hexadécimal.

Les codes-op. mnémoniques de telles instructions se terminent par la lettre Z, les trois premières lettres désignant l'opération ; ici :

ADDZ :

ADDition avec registre **Z**éro

qui réalise l'addition R0 ← (R0) + (R1), « somme des contenus des registres 0 et 1, résultat dans le registre 0 ».

Nous allons insérer cette instruction **à la place** du premier NOP du programme ; symboliquement, sur notre feuille de programmation, il vient :

On voit apparaître à chaque pression de REG la demande « r = » ; notre première frappe désignera le registre que nous voulons voir et/ou modifier : 1 fait apparaître « r1 = » ; C fait apparaître « PC = xxxx », où **PC** est, rappelons-le, l'abréviation de **P**rogram **C**ounter, le fameux **compteur ordinal** qui progresse pour désigner en mémoire les instructions successives.

La « logique » de consultation et mise à jour de ces différents éléments est la même que pour les opérations en mémoire ; par exemple, vous aurez constaté qu'après le registre 1 et sa validation par [E/N], le moniteur vous a « proposé » automatiquement le registre 2 par « r2 = ».

A ce point, nous allons solliciter, pour mieux voir le déroulement, l'exécution **pas à pas ;** mettons toutes les clés vers le bas pour entrer 0, et pressons [STEP] ; il apparaît :

0001 81 — prochain code-op. :

ADDZ, R1

La première instruction a été exécutée, ce que l'on vérifie en pressant [REG][0] et en visualisant

001	81		ADDZ	R1	R0 ← (R0) + (R1)

Pour maîtriser l'exécution d'une telle instruction, nous aurons recours au moniteur, que l'on rappellera par [MON] (si nécessaire, pressez à la fois [MON] et [RST]) ; il nous faut :

- entrer l'instruction

[MEM] [1] [E/N] [8] [1] [E/N]

- pour donner une valeur connue (prenons 1) au registre 1, utiliser la touche [REG]

[REG] [1] [1] [E/N]

- pour demander que le programme s'exécute à partir de l'adresse 0, utiliser de nouveau [REG]

[REG] [C] [0] [E/N]

« r0 = 00 » ; on vérifie par [E/N] que « r1 = 01 ». Effectuant un nouveau pas [STEP], il vient :

0002 C0 — prochain code-op. :

NOP

En visualisant de la même façon le registre 0, on voit maintenant « r0 = 01 », ce qui montre que l'addition s'est bien effectuée. Par [STEP] successifs, nous atteignons

0004 F0 — prochain code-op. :

WRTD, R0

qui est l'instruction de sortie sur les voyants ; un [STEP] supplémentaire et l'on a « 1 » sur les voyants :



Sur l'affichage, 0005 40 montre la prochaine instruction de HALTE. Un dernier [STEP] éteint RUN, et désormais, [STEP] est **inopérant.**

Cette longue description de la manipulation, étape par étape, était nécessaire pour bien voir le principe de l' « expérience » ; le lecteur peut maintenant, seul, effectuer une série d'exercices pour se familiariser avec les instructions entre registres.

Opérande « immédiat »

Pour l'opération d'addition entre registres, nous avons employé un artifice plutôt laborieux (appel au moniteur) pour injecter la valeur voulue dans le registre opérande R1. Dans ce même but, on est conduit à imaginer plutôt une instruction qui se formulerait tout bêtement : « ajouter 1 » ; instruction que l'on qualifiera facilement d'**immédiate.**

Ce type d'instruction existe dans le 2650, avec le format générique suivant :



Dans la plupart des cas, le premier octet donne, outre le code-op., le numéro d'un registre qui contiendra le **premier opérande** (avant) et le **résultat** (après) ; le deuxième octet **est** le **deuxième opérande,** et « fait partie » de l'instruction.

Par l'addition, dans ce format, le code-op. est 100001 ; le mnémonique correspondant est ADDI, comme **ADD**ition **I**mmédiate. Pour avoir le même résultat que dans notre première manipulation, nous composerons l'instruction :

|1,0,0,0,0,1|0,0| |0,0,0,0,0,0,0,1|

ADDI — R0 — valeur « 1 »

Valeur hexadécimale : 8401

que nous substituerons comme précédemment ; le « texte » du programme est modifié.

0 0 1	84 01		ADDI, R0	1	R0 — (R0) + 1
0 0 3	C0		NOP		un NOP reste

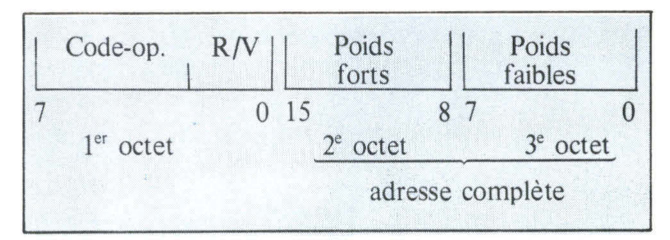
On observe cette fois-ci dans le texte qu'il y a deux octets de traduction pour une seule ligne, ce qui est la convention la plus usuelle. Cela implique, bien sûr, que l'adresse d'implantation de la ligne suivante soit en progression de 2.

Cette substitution étant effectuée dans le programme, vous pourrez vérifier que l'on obtient rigoureusement les mêmes résultats que dans la première manipulation.

Opérandes en mémoire

Avec le format immédiat, nous avons abordé une première façon, visiblement « élémentaire » d'avoir un opérande en mémoire. Un autre format d'instruction plus « absolu » s'imagine aisément, dans lequel on précise in extenso l'adresse d'un octet, qui sera l'opérande.

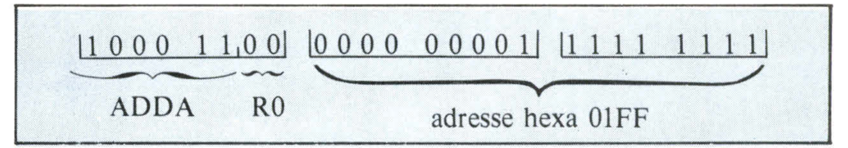
Sur le 2650, ce format est de la forme :



En fait, il est un peu compliqué par des variantes (puissantes) : l'indirection et/ou l'indexation ; ces mécanismes sont tellement importants qu'un prochain article leur sera précisément réservé, dont le sujet principal est l'organisation des données.

Cela dit, dans les cas simples, ce format est facile à interpréter. Pour obtenir (encore) un programme équivalent avec ce format d'instruction, nous trouvons dans le répertoire le code-op. **ADDA** comme **ADD**ition **A**bsolue, en binaire 1000 11.

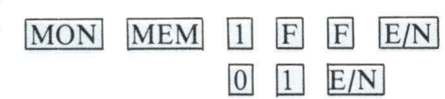
Prenons arbitrairement comme position de mémoire-opérande l'octet d'adresse hexadécimale 1FF (qui est le dernier de notre mémoire vive) ; on constitue l'instruction :



et le texte de programme se modifie comme suit :

0 0 1	8 C 0 1 F F		ADDA, R0	01FF	
0 0 4	F0		WRTD, R0		

On forcera cette nouvelle instruction entre 001 et 003, et, avant de manipuler, on peut forcer grâce au moniteur la valeur 1 à l'emplacement 1FF.



pour vérifier que le fonctionnement est à nouveau le même.

L'adressage relatif : un raccourci ingénieux

Si l'on suppose qu'un opérande se situe, en mémoire, à « quelques octets » de l'instruction, est-il bien nécessaire de se donner deux octets, pour en spécifier l'adresse ? Alors que l'on pourrait **sur un seul octet** indiquer seulement la **différence** entre l'emplacement de l'instruction (que donne le PC) et l'emplacement de l'opérande.

Sur les processeurs comme le 2650, le procédé d'adressage **relatif** permet effectivement de prendre ce « raccourci », grâce à un nouveau format d'instruction sur deux octets, ou le second octet contient le **déplacement** (en anglais : offset), c'est-à-dire la quantité qui doit s'additionner au **contenu du compteur ordinal** pour obtenir l'adresse effective de l'opérande.

Ainsi se réalise une double **économie** de **place** (un octet de moins qu'en absolu) et de **temps** (un cycle d'accès à la mémoire). Cela dit, il faut observer qu'au moment où le microprocesseur détient « toute l'instruction », le PC (par définition même) contient déjà l'**adresse de l'instruction suivante ;** autrement dit, l'adresse de l'instruction considérée, **plus 2.**

Composons par exemple l'instruction ADDR, comme **ADD**ition **R**elative, de code 1000 10, que nous allons de nouveau insérer dans notre programme par substitution, aux emplacements 001 et 002. Si l'on place (arbitrairement) l'opérande à l'adresse 20_H, le déplacement se calcule comme suit :

Emplacement instruction	001
plus 2	+ 2
adresse de référence	003

Déplacement = 020 – 003, soit 1D en hexa.

0 0 1	8 8 1 D		ADDR, R0	H' 1 D'	
0 0 3	C 0		NOP		

Le texte sera modifié comme suit (nous allons plus vite, avec l'habitude !).

A condition de placer 01 à l'emplacement 20, on trouvera, encore une fois, que le programme fonctionne à l'identique !

80 % du temps d'exécution d'un programme est passé essentiellement à recopier les informations.

Des nombres « négatifs »

Et si notre opérande avait été disposé « avant » l'instruction ? Il est clair que dans ce cas la quantité qui doit s'ajouter au contenu du PC doit être **négative.** Pour rendre compte de cette possibilité, qui existe bel et bien sur le 2650, il va nous falloir introduire la notion de nombre négatif codé en **complément à deux.**

Souvenons-nous qu'avec N bits, nous pouvons coder 2^N objets. Ainsi sur un octet, nous représentons de manière « naturelle » (voir le deuxième article) les nombres de 0 à 255 (décimal). Mais nous pouvons tout aussi bien choisir de coder tous les nombres **signés, de – 128 à + 127,** en passant par zéro ; ce qui ne demande pas plus des deux cent cinquante six combinaisons allouées.

Cela dit, ce qui nous intéresse, c'est que le code choisi soit **compatible avec l'addition binaire** ordinaire, en ce sens que la somme d'un nombre et de son opposé doit être nulle. La méthode « qui marche » est la représentation en **complément à 2,** où l'on obtient l'opposé d'un nombre en deux étapes.

(a) on **complémente** ce nombre (les 0 deviennent des 1 et vice-versa) ;
(b) on ajoute 1 au résultat.

Obtenons ainsi le complément à deux du nombre trois, sur 8 bits :

« trois »	0000 0011
(a) complémenter	1111 1100
(b) plus 1	+ 1
« moins trois »	1111 1101

Et vérifions, en bonne algèbre, que 3 + (– 3) = 0

« trois »	0000 0011
« moins trois »	+ 1111 1101
donne « zéro »	0000 0000

ce qui est le résultat voulu (si l'on ne tient pas compte de la dernière retenue, qui va au-delà des 8 bits).

On constatera facilement que dans cette convention le bit le plus à gauche est à « 1 » pour les nombres négatifs, et à « 0 » pour les nombres positifs (ou le zéro). De plus, jusqu'à 127 (décimal), soit 7F (hexa), les nombres positifs sont codés de manière « naturelle ».

A retenir aussi deux nombres « remarquables » dans la convention de complément à 2 : 1111 1111 (ou « plein 1 ») qui représente « – 1 » ; et 1000 0000 qui représente « – 128 ». En hexadécimal sur un octet : FF et 80.

Vision globale du répertoire

Ouf ! Nous avons fini avec les préliminaires indispensables. En effet, pour le microprocesseur 2650, la lecture du répertoire est d'une simplicité (presque) enfantine, dès l'instant où l'on sait que toutes les instructions, à peu de chose près, sont construites selon l'un des quatre modèles déjà vus.

De plus, le fabricant est bon avec nous, car il observe **des règles strictes pour les noms mnémoniques** de ses instructions :

- les opérations entre le registre 0 et un autre registre s'identifient par la lettre Z (exemple : LOD**Z**) ;
- les opérations entre un registre et un opérande immédiat, par la lettre I (exemple : EOR**I**) ;
- les opérations avec la mémoire, par la lettre A si l'adresse est absolue (exemple : COM**A**), ou la lettre R si elle est relative (exemple : SUB**R**).

Nous sommes en mesure à ce niveau d'expliquer les tableaux de la **figure 5** dans lequel sont regroupés **toutes les instructions** du 2650.

Le tableau opérations regroupe toutes les instructions intuitivement « **opératoires »,** avec quatre colonnes pour les quatre registres possibles. En regard de chaque code-op. mnémonique, la traduction en hexadécimal est donnée directement, elle incorpore le numéro de registre.

Le tableau branchements résume l'impressionnant arsenal d'instructions de **rupture de séquence,** donc de **décision** dont dispose le 2650.

Le tableau entrée/sortie donne les 6 instructions d'échange d'information « avec l'extérieur ». On notera que **chacun des quatre registres** peut, indifféremment, s'employer comme source ou destination des octets échangés.

Le tableau mot d'état donne dix instructions, qui se réfèrent à deux registres très « spéciaux » du 2650 (PSU et PSL) dont nous n'avons pas encore parlé. Enfin, on a deux petits tableaux pour des opérations spéciales (et qui portent bien leur nom) et les divers, que nous connaissons déjà.

Exercice :

Repérez dans ces divers tableaux les instructions dont nous nous sommes déjà servis.

Les instructions de recopie

Si vous avez déjà programmé en langage-machine, vous le savez ; sinon, il vous faut savoir (et vous vérifierez bien vite) que, dans un programme, on passe l'essentiel de temps à **recopier** l'information. Dans le « meilleur » des cas, 10 à 20 % au plus de l'activité se consacre au « calcul » !

A **l'intérieur** du micro-ordinateur, les Anglo-Saxons ont pris plus ou moins l'habitude d'établir une **hiérarchie** entre les **sources** et **destinations** possibles, à savoir, **entre les registres** généraux, et **entre ces registres et la mémoire.** Selon cette hiérarchie, on « charge » **(LOAD)** ou on « range » **(STORE)** les informations.

Pour le 2650, un petit diagramme **(fig. 6)** permet de retenir cette hiérarchie, et la signification des différentes opérations.

Les entrées/sorties, échanges avec l'extérieur, s'expriment, elles, comme des « lectures » **(READ)** et des « écritures » **(WRITE),** d'un **port d'entrée** (input-port) vers un registre, et d'un registre vers un **port de sortie** (output-port).

Ⓐ

Opérations	R0	R1	R2	R3
LODZ	00	01	02	03
LODI	04	05	06	07
LODR	08	09	0A	0B
LODA	0C	0D	0E	0F
STRZ		C1	C2	C3
STRR	C8	C9	CA	CB
STRA	CC	CD	CE	CF
ADDZ	80	81	82	83
ADDI	84	85	86	87
ADDR	88	89	8A	8B
ADDA	8C	8D	8E	8F
SUBZ	A0	A1	A2	A3
SUBI	A4	A5	A6	A7
SUBR	A8	A9	AA	AB
SUBA	AC	AD	AE	AF
ANDZ	—	41	42	43
ANDI	44	45	46	47
ANDR	48	49	4A	4B
ANDA	4C	4D	4E	4F
IORZ	60	61	62	63
IORI	64	65	66	67
IORR	68	69	6A	6B
IORA	6C	6D	6E	6F
EORZ	20	21	22	23
EORI	24	25	26	27
EORR	28	29	2A	2B
EORA	2C	2D	2E	2F
COMZ	E0	E1	E2	E3
COMI	E4	E5	E6	E7
COMR	E8	E9	EA	EB
COMA	EC	ED	EE	EF
RRL	D0	D1	D2	D3
RRR	50	51	52	53

Branchements	N.C.	R0	R1	R2	R3	< NEG	⩽	= NUL	⩾	> POS	=
ZBRR	9B										
BCTR	1B					1A		18		19	
BCFR							99		9A		98
BCTA	1F					1E		1C		1D	
BCFA							9D		9E		9C
BIRR		D8	D9	DA	DB						
BIRA		DC	DD	DE	DF						
BDRR		F8	F9	FA	FB						
BDRA		FC	FD	FE	FF						
BRNR		58	59	5A	5B						
BRNA		5C	5D	5E	5F						
BXA	9F										
ZBSR	BB										
BSTR	3B					3A		38		39	
BSFR							B9		BA		B8
BSTA	3F					3E		3C		3D	
BSFA							BD		BE		BC
BSNR		78	79	7A	7B						
BSNA		7C	7D	7E	7F						
BSXA	BF										
RETC	17					16		14		15	
RETE	37					36		34		35	

Ⓑ

Fig. 5. – Le jeu d'instruction du 2650 est classé en six groupes :

A - Les opérations.
B - Les branchements.
C - Les entrées/sorties.
D - Les instructions spéciales.
E - Les instructions concernant le mot d'état.
F - Les diverses : NOP et HALT.

Ⓒ

Entrée/ Sortie	R0	R1	R2	R3
REDD	70	71	72	73
REDC	30	31	32	33
REDE	54	55	56	57
WRTD	F0	F1	F2	F3
WRTC	B0	B1	B2	B3
WRTE	D4	D5	D6	D7

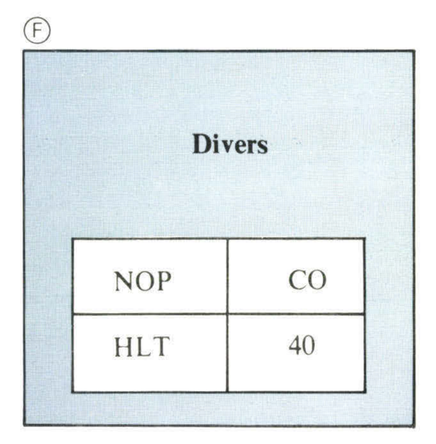
Ⓓ

Spéciales	R0	R1.	R2	R3
TMI	F4	F5	F6	F7
DAR	94	95	96	97

Ⓔ

Mot d'état	
LPSU	92
LPSL	93
SPSU	12
SPSL	13
PPSU	76
PPSL	77
CPSU	74
CPSL	75
TPSU	B4
TPSL	B5

Ⓕ



Divers

NOP	CO
HLT	40

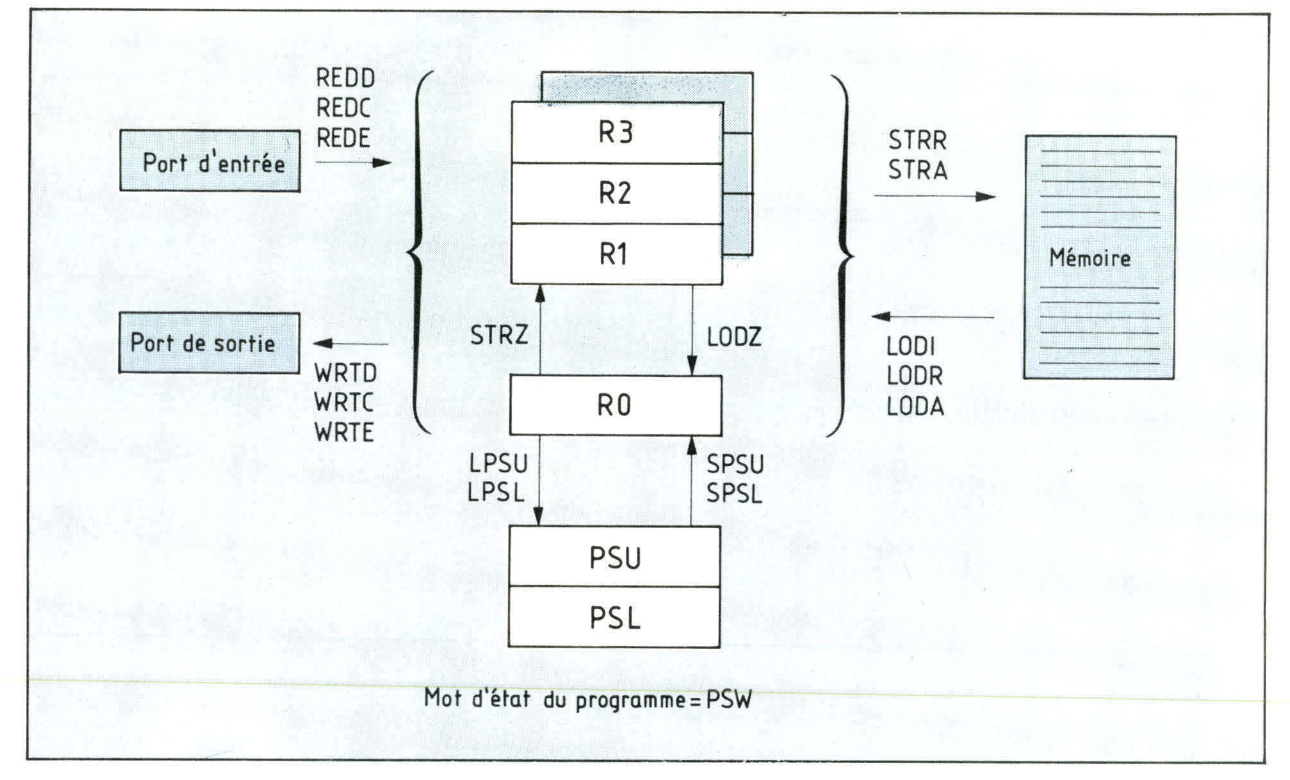


Fig. 6. – Différents échanges et transferts d'informations et leurs mnémoniques, entre les registres, la mémoire et les ports d'entrée/sortie.

Les opérations arithmétiques et logiques

Comme la plupart des microprocesseurs de la même classe, le 2650 n'a qu'un petit nombre de telles opérations ; cependant, ces opérations sont suffisantes pour que, par **combinaison** et **répétition,** on puisse effectuer des multiplications, divisions, etc.

Les additions et soustractions **(ADD** et **SUB)** se passent de commentaires. Les opérations logiques disponibles sont le ET **(AND),** le OU inclusif **(IOR)** et le OU exclusif **(EOR) ;** des tables de vérité bien connues donnent leur résultat, chaque opération logique s'effectuant « bit à bit » entre deux octets.

AND	0	1
0	0	0
1	0	1

IOR	0	1
0	0	1
1	1	1

EOR	0	1
0	0	1
1	1	0

(différence)

Les opérations de comparaison **(COM)** seront traitées plus naturellement avec les branchements, dans le processus de prise de décision.

Nous restent les deux opérations de décalage, ou **rotation,** qui agissent sur un seul opérande, dans un registre, par recopie des bits d'un rang vers la gauche **(RRL, R**otate **R**egister **L**eft) ou vers la droite **(RRR, R**otate **R**egister **R**ight). Avec quelques précautions, ces deux instructions peuvent s'interpréter comme une multiplication (x 2, pour RRL) et une division (:2, pour RRR) rudimentaires.

Commentaires et perspectives

Au point où nous en sommes, nous avons déjà **une bonne perception des opérations** de notre micro-ordinateur ; mais nous avons **laissé de côté :**

- les **mécanismes de décision et d'organisation** (branchements, sous-programmes) ;
- les **variantes puissantes de l'adressage** (indirection et indexation).

La raison est que les considérations sur la **structure des programmes** et sur la **structure des données** sont si importantes, qu'elles nous ont paru justifier **un article entier,** sur chaque thème.

Arrivé au troisième numéro, le lecteur peut être dans l'une ou l'autre de deux situations :

- Il a fait l'acquisition d'un Instructeur 50. Il est certain, dans ce cas, qu'il aura commencé à fabriquer des programmes beaucoup plus copieux que ceux de notre initiation. Alors, nous pensons que la comparaison entre ses expériences et nos prochains articles sera fructueuse, car nous pénétrerons sur le terrain soi-disant « avancé » de la **programmation structurée.**
- Il n'a pas (ou pas encore) ce matériel. Alors, qu'il considère que la suite de notre série lui permettra, le jour venu, d'éviter le plus possible de ces « faux-pas » qui sont le lot de tous les programmeurs, professionnels ou non ! ■

Jean-Michel COUR

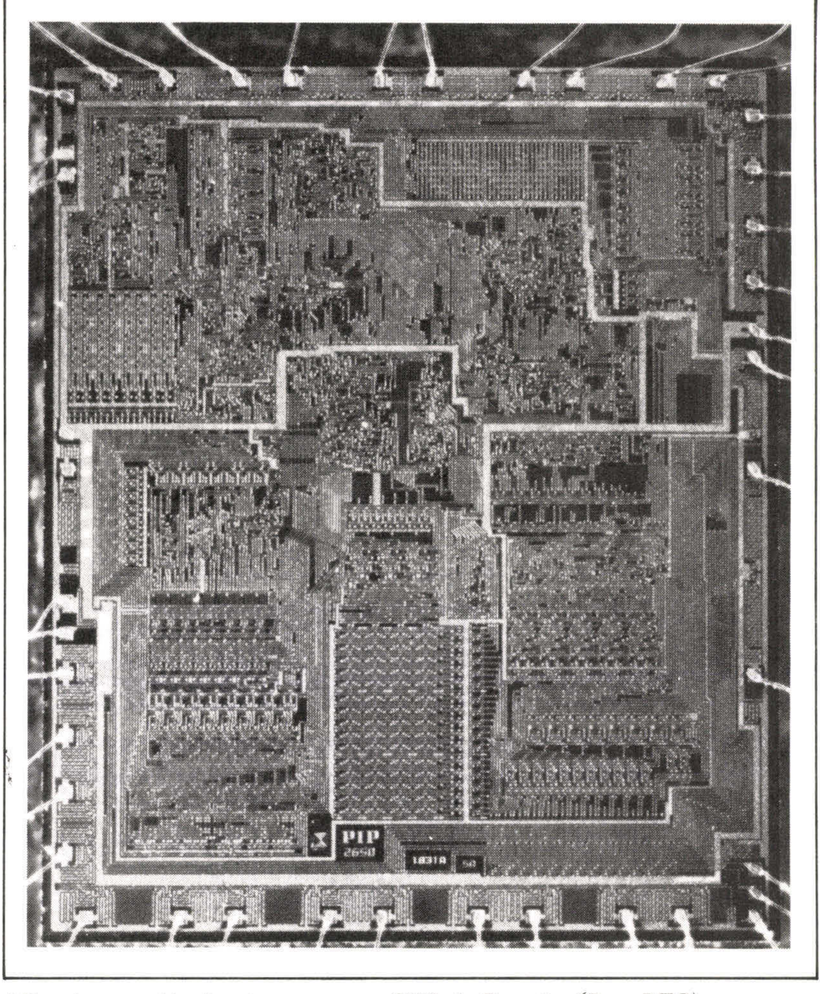
* *Jean-Michel COUR anime le groupe « Architecture Informatique » spécialisé dans l'ingénierie micro-informatique au sein de la société CISI.*

L'unité de commande

L'Unité Arithmétique et Logique (ALU) a pour but essentiel, le traitement, la manipulation et la gestion des données. Elle peut réaliser des fonctions logiques telles que ET, OU, inversion..., des fonctions arithmétiques comme l'addition, la soustraction, l'incrémentation et la décrémentation ainsi que des fonctions de comparaison ou de test.

Les informations nécessaires au fonctionnement de l'ALU sont stockées en mémoire et représentent deux types bien distincts : les données ou informations traitées (opérandes) appliquées sur les entrées de l'Unité Arithmétique et Logique seront traitées par elle ; et les instructions du programme (informations traitantes) ou code opération seront analysées par l'unité de commande aussi appelée unité de contrôle.

Ainsi, l'unité de commande, que nous étudions aujourd'hui, est l'organe du microprocesseur qui gère les instructions que le programmeur a inscrit en mémoire. Pour cela, elle est constituée d'un registre d'instruction qui stocke les instructions issues de l'emplacement mémoire adressé par le compteur de programme ; d'un décodeur nécessaire au décodage du mot contenu dans le registre d'instruction et d'un séquenceur central qui génère les micro-instructions indispensables au bon cheminement des informations.



Microphotographie du microprocesseur 2650 de Signetics (Doc. RTC).

Il n'est bien entendu pas possible, au niveau de cet article, d'étudier de façon détaillée les problèmes posés par la conception et l'optimalisation de l'unité de commande. Nous nous contenterons d'une approche intuitive du problème en choisissant le séquencement d'une instruction que nous avons déjà étudié lors d'un précédent article * :

LDA A (adresse)

La **figure 1** indique l'emplacement de l'unité de commande dans un système à microprocesseur. Le schéma synoptique de cette unité de commande est donné **figure 2.**

Elle comporte les principales parties suivantes :

— un registre d'instruction
— un décodeur
— un séquenceur central.

Dans les microprocesseurs, le séquenceur est synchrone, c'est-à-dire que les micro-instructions seront émises au rythme de la fréquence d'horloge.

** Micro-Systèmes numéro 2, page 85 : « Le cheminement des informations dans un microprocesseur ».*

Notions de séquenceur central

Le séquenceur central est l'organe qui génère les micro-instructions nécessaires au bon cheminement des informations. Les valeurs d'entrées du séquenceur sont fournies d'une part par l'instruction et les conditions d'adressage, d'autre part, par l'état de la machine qui regroupe un certain nombre d'informations telles que les codes conditions de l'Unité Arithmétique et Logique et les demandes d'interruptions, états de bouton du pupitre de commande...

Décodage de l'instruction

L'instruction est décodée à l'aide d'un décodeur digital, circuit combinatoire à l'entrée duquel on applique un code à n bits. La sortie est constituée de N sorties, avec $N = 2^n$.

Une seule sortie se trouve à 1 pour chacune des valeurs du code d'entrée. Ce type de décodeur, appelé « 1 parmi N » est réalisé avec des circuits combinatoires. Traçons la table de vérité et les équations logiques correspondantes dans le cas ou N est égal à 4.

Le séquenceur central est l'organe qui génère les micro-instructions nécessaires au cheminement des informations.

CODE ENTREE		SORTIES				
A	B	S_0	S_1	S_2	S_3	Equations logiques des sorties à « 1 »
0	0	1	0	0	0	$S_0 = \overline{A} . \overline{B}$
1	0	0	1	0	0	$S_1 = A . \overline{B}$
0	1	0	0	1	0	$S_2 = \overline{A} . B$
1	1	0	0	0	1	$S_3 = A . B$

Le schéma logique de cette table de vérité est donné **figure 3.** Bien sûr, si le code d'entrée est à 8 bits, on peut de cette façon réaliser un décodeur à « 1 parmi 256 ». Ce décodeur nécessiterait beaucoup de portes logiques aussi préfère-t-on utiliser d'autres solutions pour réaliser le décodage. Etudions les différentes possibilités que nous pouvons envisager de réaliser.

Principe du décodeur à mémoire morte

Les premiers décodeurs à composants discrets étaient conçus autour d'une matrice à diodes **(fig. 4).**

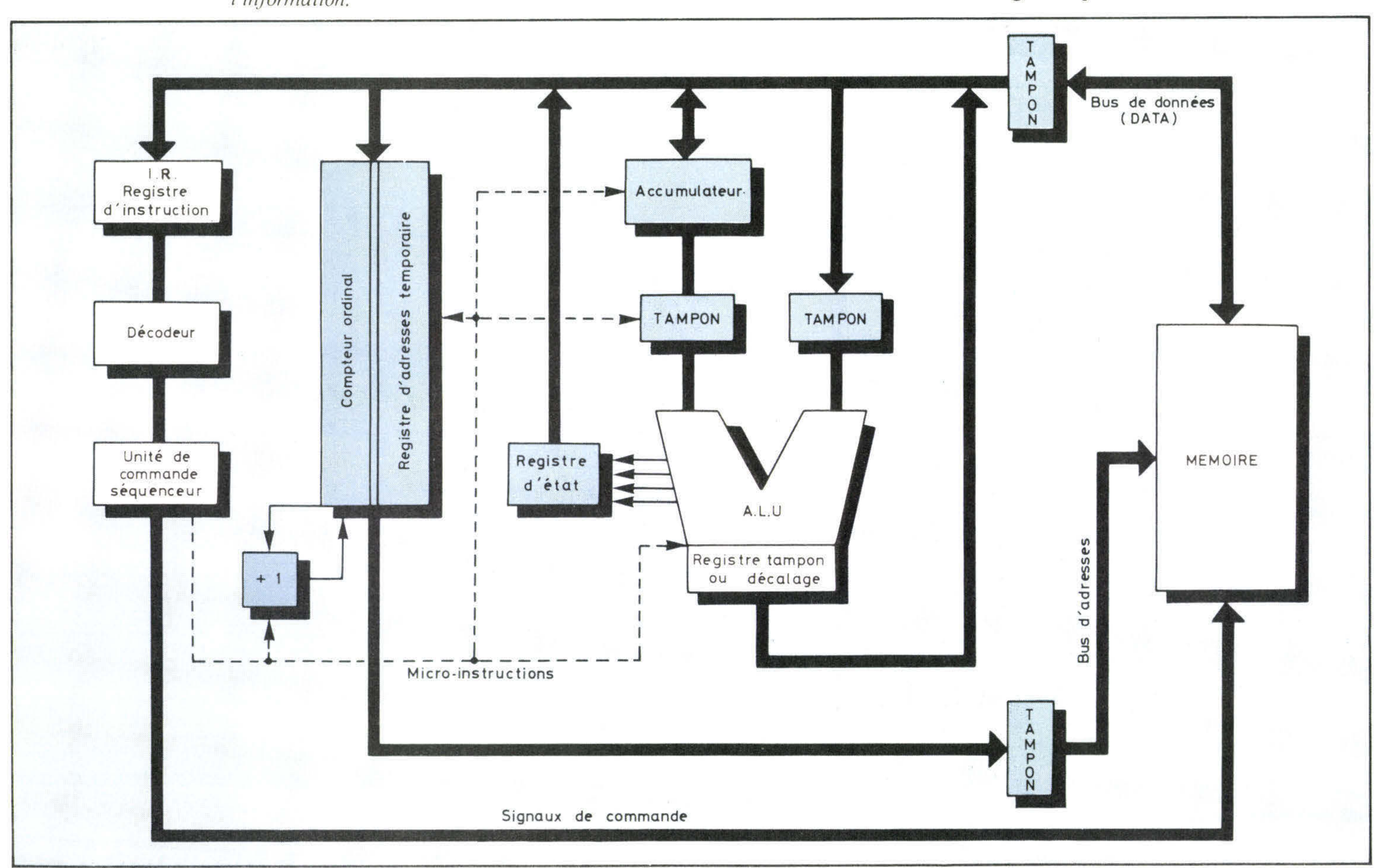
Prenons l'exemple du mot 1001, correspondant à des tensions de + 5,0, 0, + 5 V. Les sorties des lignes de décodage 1,3 et 4 afficheront 0 V, car l'une ou l'autre des diodes conduisent, faisant circuler un courant dans la résistance R, la diode et la masse. La ligne 2 de décodage restera au niveau haut, + 5 V, avec les deux diodes bloquées. En effet, ces diodes reçoivent d'une part le + 5 V par la résistance R et d'autre part un niveau haut + 5 V par le code d'entrée.

Malheureusement, un code 1101 ou 1011, 1111 sera lui aussi décodé. Pour éviter ce genre d'incertitudes, on est amené à utiliser des inverseurs et le décodage simultané du code et de son inverse.

Le décodage est dans ce cas parfaitement défini et correspond au code : $b_0, \overline{b_1}, \overline{b_2}, b_3$ **(fig. 5).**

Or dans la technologie MOS, il est plus facile d'intégrer des transistors que des diodes. En conséquence, le même montage, compte tenu des règles simples, pour un transistor MOS canal N (+ 1 sur la grille produit la conduction du

Fig. 1. – Emplacement de l'unité de commande dans un système de traitement de l'information.



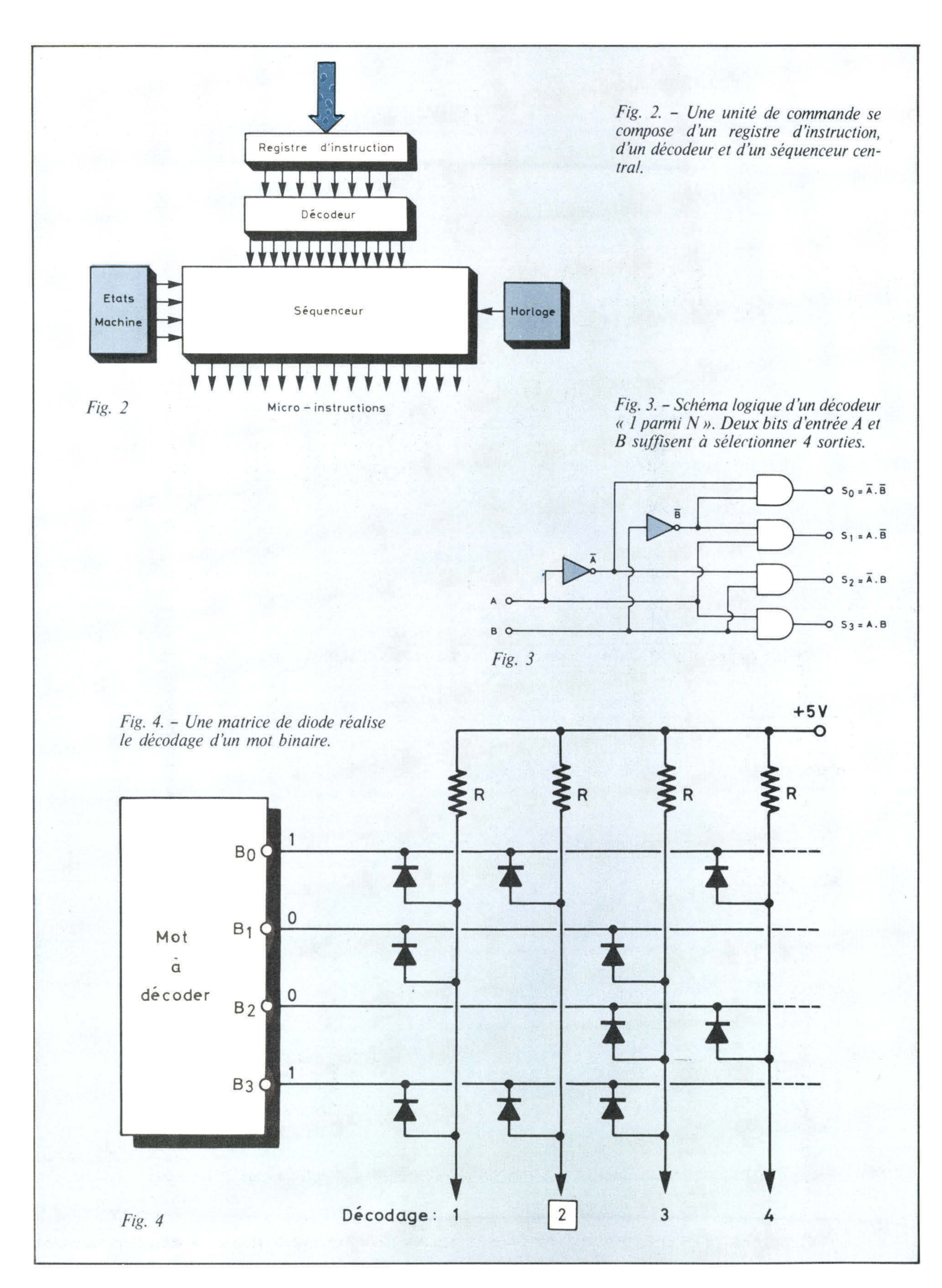


Fig. 2

Fig. 2. – Une unité de commande se compose d'un registre d'instruction, d'un décodeur et d'un séquenceur central.

Fig. 3

Fig. 3. – Schéma logique d'un décodeur « 1 parmi N ». Deux bits d'entrée A et B suffisent à sélectionner 4 sorties.

Fig. 4

Fig. 4. – Une matrice de diode réalise le décodage d'un mot binaire.

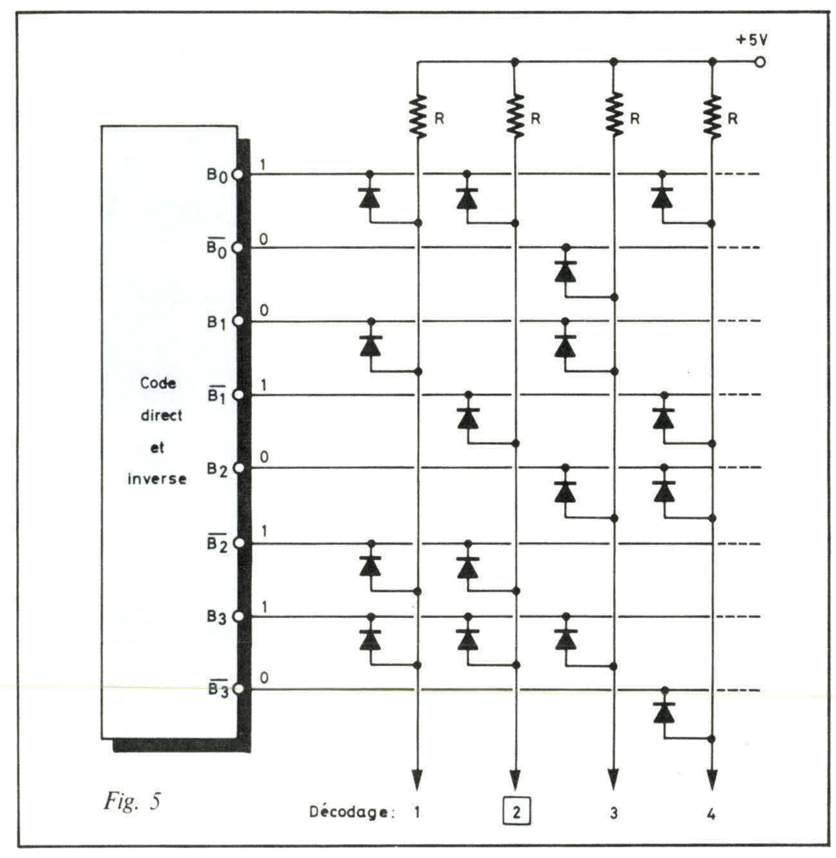


Fig. 5

Fig. 5. – Pour définir sans ambiguité la sortie sélectionnée, il est nécessaire de décoder simultanément le code binaire et son inverse. Dans notre cas, la sortie 2 est sélectée lorsque le code 1001 est présent sur b_0, b_1, b_2 et b_3.

Fig. 6. – Décodeur « 1 parmi N » réalisé en technologie MOS à canal N. Les diodes sont maintenant remplacées par des MOS.

Fig. 7. – L'épaisseur de la couche d'oxyde d'un transistor MOS définit son état. Lorsque la couche est épaisse, le déblocage du transistor est impossible, son état est « 0 ».

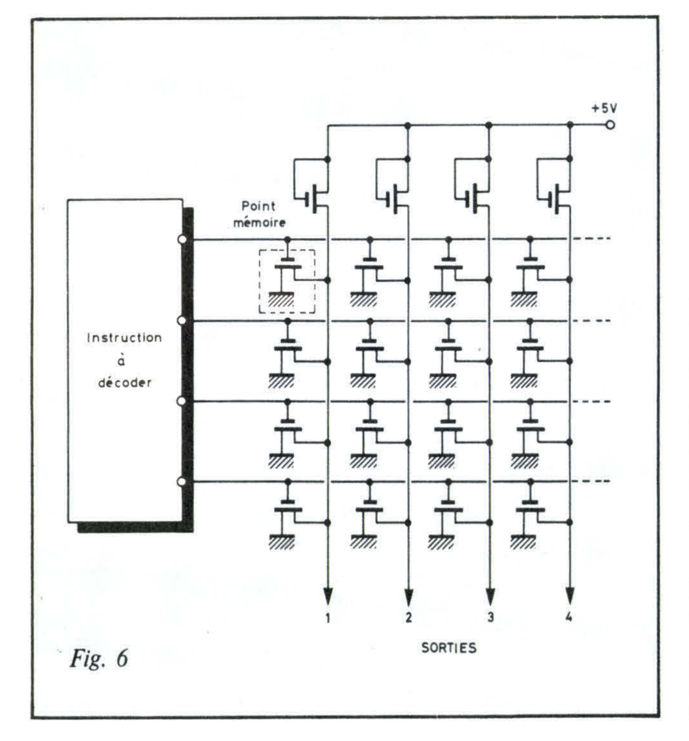


Fig. 6

transistor, sinon il est bloqué) permet de tracer le schéma de la **figure 6.**

Décodeur à mémoire morte

Une matrice de décodage est un quadrillage parfait où les colonnes sont alimentées en permanence, les rangées étant connectées par l'intermédiaire de l'instruction à décoder.

Le point mémoire est constitué par un transistor MOS commandé par la « gate ». Suivant l'épaisseur de la couche d'oxyde sous la grille, on obtient facilement un « 1 » ou un « 0 » comme le montre le schéma de la **figure 7.** Certains MOS possèdent une couche d'oxyde mince (état 1) alors que d'autres ont la métallisation « gate » déposée sur une couche d'oxyde épais (état 0) et rend ainsi leur déblocage impossible.

Analysons à titre d'exemple le décodage de l'instruction LDA,A (adresse) qui charge le contenu de la mémoire désignée par l'adresse dans l'accumulateur. Le code de cette instruction est :

$$\underbrace{1\,0\,1\,1}_{B}\quad\underbrace{0\,1\,1\,0}_{6}$$

en hexadécimal

Le décodage de cette instruction est réalisé par l'équation logique suivante :

$$\text{LDA A} = D_7\,\overline{D_6}\,D_5\,D_4\,\overline{D_3}\,D_2\,D_1\,\overline{D_0}$$

La matrice de décodage étant constituée par des fonctions OU, nous pouvons transformer cette expression d'après le théorème de

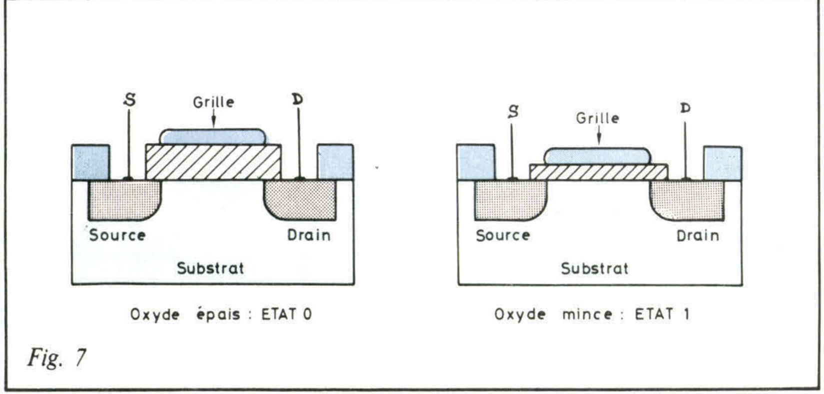


Fig. 7

** Théorème de MORGAN : $X = \overline{a \cdot b} = \overline{a} + \overline{b}$*

Morgan * de façon à ne faire apparaître que des opérations OU logique.

Ainsi l'équation de LDA A devient :

$$\overline{LDA\ A} = \overline{D_7 \overline{D_6} D_5 D_4 \overline{D_3} D_2 D_1 \overline{D_0}}$$
$$= \overline{D_7} + D_6 + \overline{D_5} + \overline{D_4} + D_3 + \overline{D_2} + \overline{D_1} + D_0$$
$$(1)\ \overline{LDA\ A} = (\overline{D_7} + \overline{D_5} + \overline{D_4} + \overline{D_2} + \overline{D_1}) + (D_6 + D_3 + D_0)$$

Le schéma logique de $\overline{LDA\ A}$ est représenté **figure 8** (les points mémoires « 0 » sont en couleur). Il est aisé de voir que l'équation est vérifiée quand D_7 est à 0 ($\overline{D_7}$) ou lorsque D_6 est à 1 (D_6) ou encore si D_5 est à 0 ($\overline{D_5}$) et nous retrouvons la relation initiale (1).

Fig. 8. - Décodage de l'instruction LDA A. Les transistors MOS en couleur sont à l'état « 0 ».

Décodeur en structure P.L.A.

P.L.A. provient de l'anglais **Pro**grammable **L**ogic **A**rray ou reseau logique programmable. Le principe d'un réseau logique programmable est basé sur la remarque suivante : **toute fonction logique peut s'écrire sous forme d'une somme de produit.**

En effet, si nous reprenons notre exemple précédent, l'équation.

$$LDA\ A = D_7 \overline{D_6} D_5 D_4 \overline{D_3} D_2 D_1 \overline{D_0}$$

peut se transformer en une somme de deux produits.

$$LDA\ A = (D_7 D_5 D_4 D_2 D_1) \cdot (\overline{D_6}\,\overline{D_3}\,\overline{D_0})$$

d'où :

$$\overline{LDA\ A} = \overline{(D_7 D_5 D_4 D_2 D_1) \cdot (\overline{D_6}\,\overline{D_3}\,\overline{D_0})}$$
$$\overline{LDA\ A} = (\overline{D_7 D_5 D_4 D_2 D_1}) + (\overline{\overline{D_6}\,\overline{D_3}\,\overline{D_0}})$$

Fig. 9. - Schéma de principe d'un réseau logique programmable (PLA).

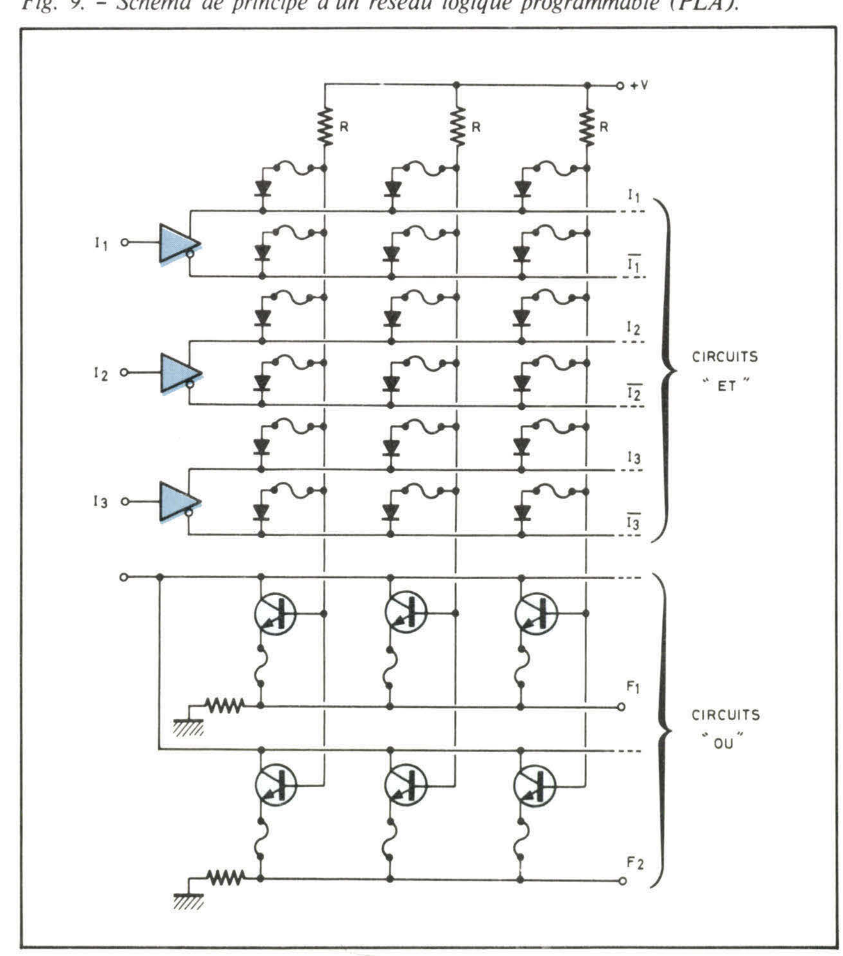
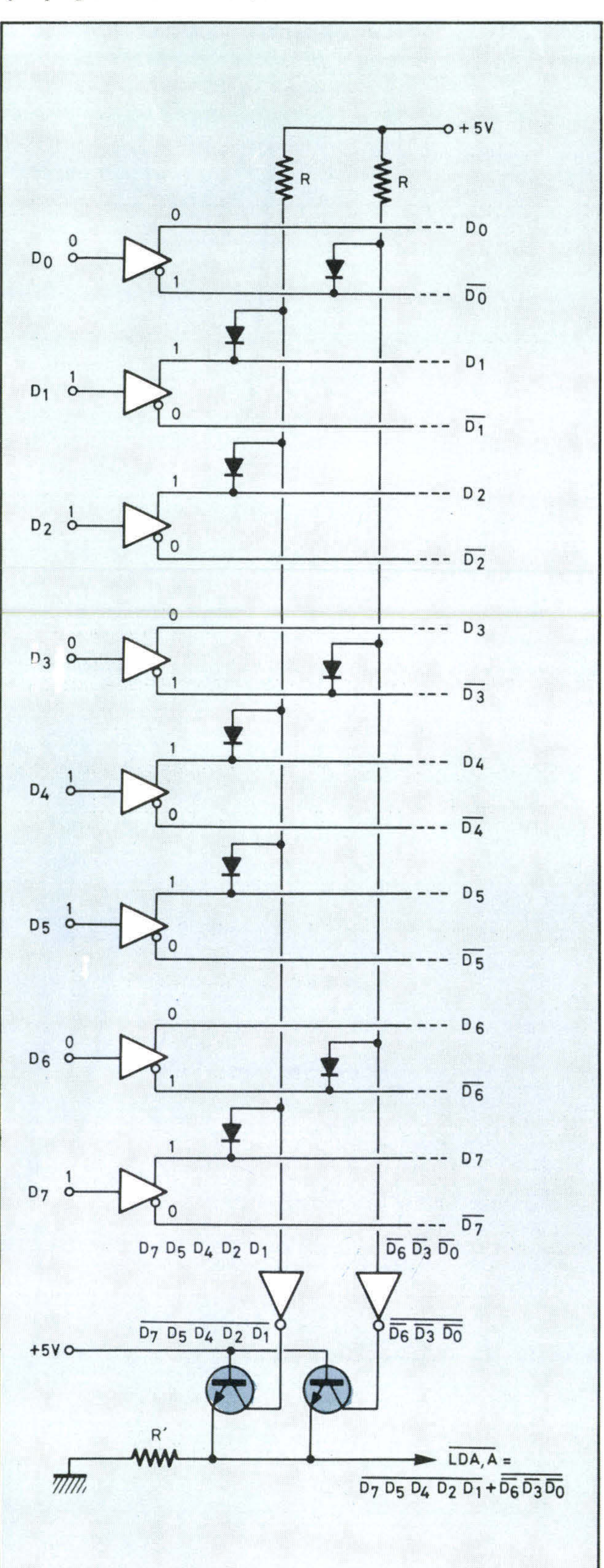


Fig. 10. – Décodage de l'instruction LDA A avec un réseau logique programmable à diodes.



Cette façon d'écrire l'équation logique, permet d'utiliser un réseau de logique programmable (P.L.A.).

Nous avons ici une relation équivalente à une somme de produit. Il est alors possible de traduire cette équation par un schéma logique groupant une matrice « produits » **(circuits ET)** et une matrice « sommes » **(circuits OU).** Le schéma de principe d'un P.L.A. est donné **figure 9.**

Les portes ET sont réalisées à l'aide d'un réseau à diodes programmables par l'utilisateur selon la technologie à fusible.

A chaque entrée se trouve un inverseur pour avoir la variable et son complément ce qui est très utile dans l'écriture des équations logiques.

Les portes OU sont réalisées par des transistors montés en émetteurs suiveurs.

Si l'on a par exemple un P.L.A. à 16 entrées et 8 sorties, il est possible de réaliser $2^{16} \times 8 \simeq 512$ k combinaisons. Ceci est donc équivalent à une mémoire ROM à 512 K points-mémoires ou de 524 288 bits.

Un P.L.A. à capacité égale occupe donc moins de surface de silicium qu'une ROM. C'est là son principal intérêt.

Réalisation de LDA A

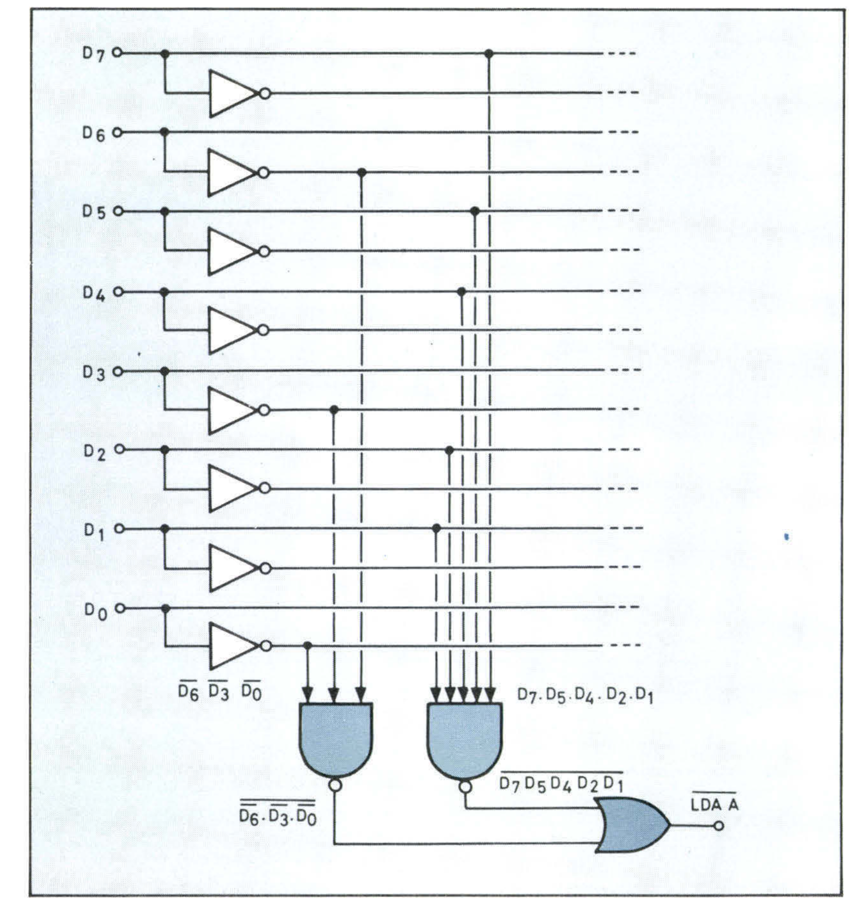
Au lieu d'utiliser des diodes **(fig. 10),** il est possible d'utiliser des transistors MOS. Dans ce cas, la matrice « Produits » est constituée de portes NAND.

Le schéma de principe du décodage de l'instruction LDA A est alors celui de la **figure 11.**

Réalisation du séquenceur

L'exécution d'une instruction exige un certain nombre de micro-instructions élémentaires.

Fig. 11. – Principe d'un décodage réalisé à l'aide de portes NAND et OR.



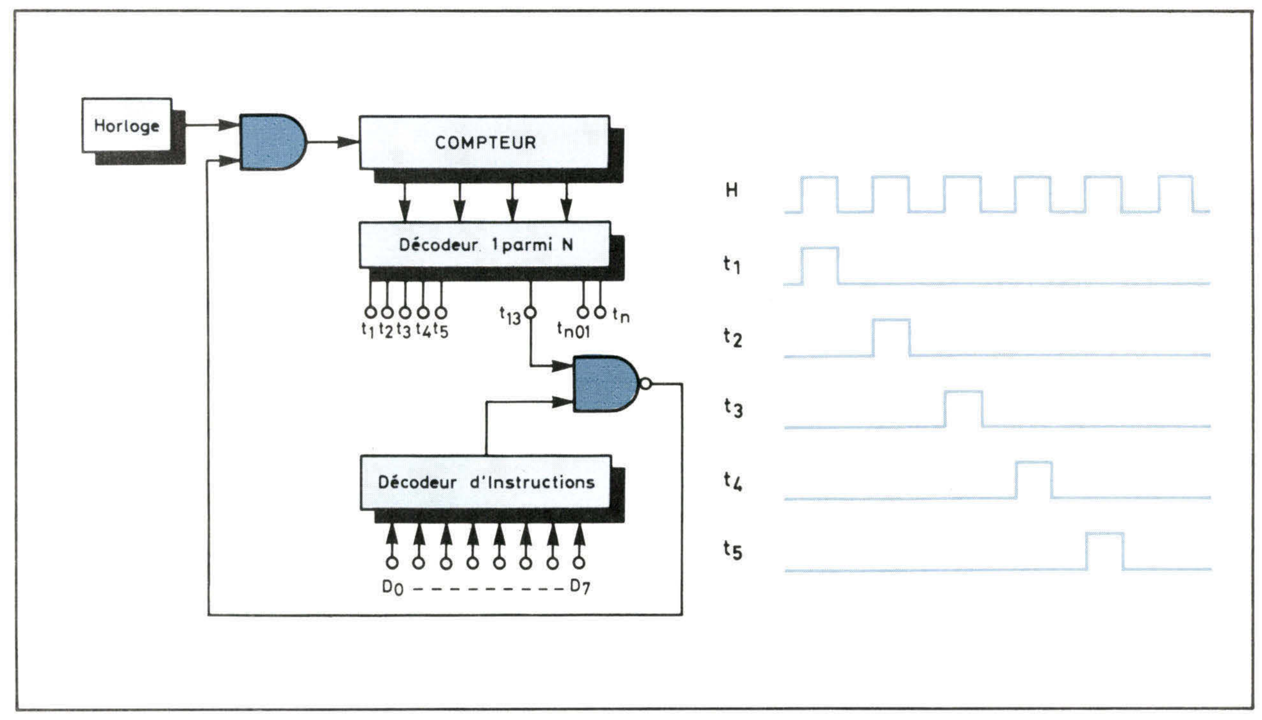


Fig. 12. – Dans un distributeur de phase, les impulsions d'horloge incrémentent un compteur. Un décodeur « 1 parmi N » fournit successivement sur chacune de ses sorties les phases t_1, t_2 ... t_n.

Il est très important que chacune de ces micro-instructions, générée par le séquenceur, soit repérée dans le temps : par conséquent, leurs phases doivent être émises par un distributeur de phase. Le principe de ce distributeur de phase est représenté **figure 12.**

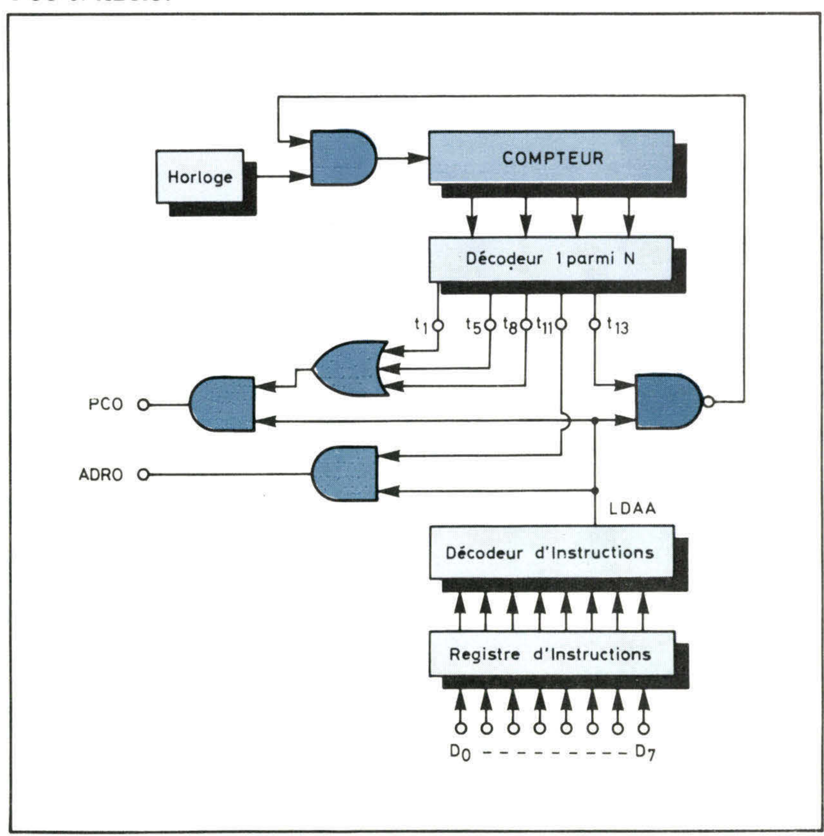
Des impulsions d'horloge sont appliquées à un compteur. Un décodeur du type à 1 parmi N fournit successivement à chacune de ses sorties, les phases t_1, t_2, t_3..., t_n.

Reprenons l'exemple de l'instruction LDA A, en appliquant l'information à la sortie du décodeur d'instruction et la phase t_{13}, sur un circuit NAND, le distributeur de phases s'arrêtera automatiquement à la treizième impulsion d'horloge.

Equations logiques des micro-instructions

Il faut maintenant écrire l'équation logique de chacune des micro-instructions. A titre d'exemple, nous pouvons établir pour l'instruction LDA l'équation de la micro-instruction PCO (mettre l'information du compteur de programme sur le bus d'adresse) qui

Fig. 13. – Schéma logique d'un séquenceur appliqué au décodage des instructions PCO et ADRO.



La micro-programmation consiste à stocker en mémoire un programme qui génère les micro-commandes dans l'ordre nécessaire à l'exécution de l'instruction.

est générée durant les temps t_1, t_5 et t_8. L'équation de PCO s'écrit :

$$PCO = LDA\ (t_1 + t_5 + t_8)$$

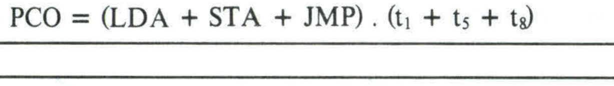
De même, si nous prenons la micro-instruction ADRO (expédier le contenu du registre d'adresse temporaire sur le bus adresse), qui doit être présenté au temps t_{11}, nous aurons :

$$ADRO = LDA\ .\ t_{11}$$

Il est donc facile de réaliser le schéma logique du séquenceur pour ces deux micro-instructions **(fig. 13).** Bien sûr, il faut étudier chacune des micro-instructions pour l'ensemble des instructions. Par exemple, si nous regardons l'équation de PCO pour les instructions LDA (adresse), STA (adresse) et JUMP (adresse), nous aurons :

$$PCO = LDA\ (t_1 + t_5 + t_8)$$
$$PCO = STA\ (t_1 + t_5 + t_8)$$
$$PCO = JMP\ (t_1 + t_5 + t_8)$$

PCO peut alors s'écrire :



$$PCO = (LDA + STA + JMP)\ .\ (t_1 + t_5 + t_8)$$

Inconvénients de la logique câblée

Cette approche élémentaire nous montre que pour l'ensemble des instructions, les équations booléennes des micro-instructions, seront plus ou moins complexes et nécessiteront surtout de nombreux circuits de logiques combinatoires.

Aussi, en pratique, ces circuits sont remplacés par un réseau logique programmable (P.L.A.).

En effet, l'instruction PCO peut se transformer comme suit :

$$PCO = (LDA + STA + JMP)\ (t_1 + t_5 + t_8)$$

et

$$\overline{PCO} = \overline{(LDA + STA + JMP)\ (t_1 + t_5 + t_8)}$$
$$\overline{PCO} = \overline{(LDA + STA + JMP)} + \overline{(t_1 + t_5 + t_8)}$$
$$= \overline{LDA}\ .\ \overline{STA}\ .\ \overline{JMP} + \overline{t_1}\ .\ \overline{t_5}\ .\ \overline{t_8}$$

ce qui est facile à réaliser avec une structure P.L.A.

Ces transformations d'équations se feront de même pour toutes les micro-instructions. L'ensemble des circuits de décodage d'instruction, distributeur de phases, et codage des équations de micro-instructions, seront gérés à l'aide d'un P.L.A.

Notions de micro-programmation

L'évolution normale de la technologie permettra d'obtenir des vitesses de fonctionnement plus élevées et des densités d'intégration plus grandes.

Ainsi, dans l'avenir, des instructions de plus en plus complexes seront à la disposition de l'utilisateur. Par exemple dans le microprocesseur TMS 9940 de Texas Instrument, on trouve l'instruction de **multiplication de deux nombres.** On dit alors que ces instructions sont microprogrammées.

La micro-programmation consiste donc à stocker en mémoire un programme qui génère les **micro-commandes** dans l'ordre, nécessaires à l'exécution de l'instruction. Les bases de la microprogrammation ont été définies par Wilkes en 1951 d'après le modèle de la **figure 14.**

L'instruction d'entrée est considérée comme la première micro-instruction du programme. Elle est décodée, et par l'intermédiaire d'une matrice (mémoire morte, ou PLA en général), elle fournit les micro-commandes nécessaires aux registres et à l'A.L.U.

En même temps, elle génère l'adresse de la micro-instruction suivante qui sera appliquée au décodeur, au prochain cycle d'horloge.

Le codage des micro-instructions

Il existe deux principes possibles de codage :

● A chaque bit de la micro-instruction est associé une micro-commande. Cette méthode entraîne une longueur exagérée de la micro-instruction, compte tenu du nom-

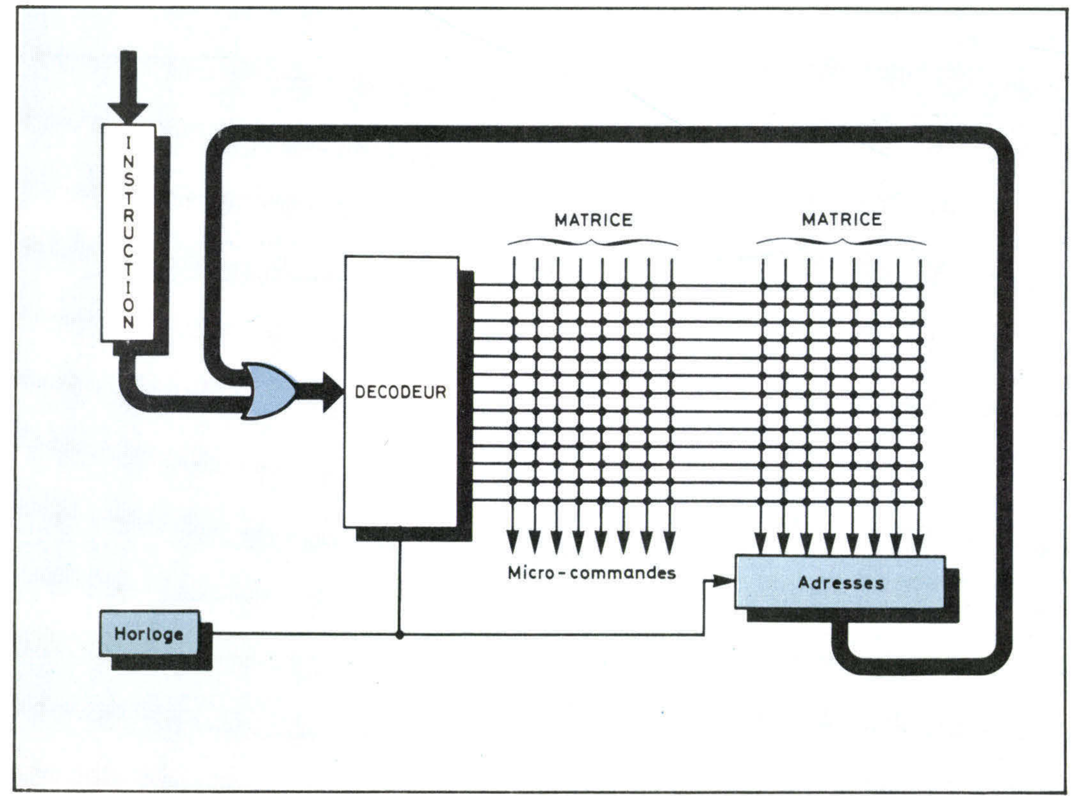


Fig. 14. - Modèle de Wilkes définissant les bases de la microprogrammation.

bre important de micro-commandes nécessaires au bon fonctionnement du système.

- A chaque micro-commande correspond une configuration binaire codée de la micro-instruction. Cette méthode nécessite un décodeur du type 1 parmi N utilisant un grand nombre de transistors.

En pratique, on utilise un compromis entre ces deux méthodes que l'on appelle **le codage par champs.** Les différentes micro-commandes sont divisées en groupes indépendants fonctionnels (commande de portes sur l'A.L.U., commande 3e état sur registre, commandes de chargement, déchargement, décalage, incrémentation...). Pour cela, on associe un « codage de champ » dans la micro-instruction.

A l'intérieur d'un champ, à chaque bit est associée une micro-commande **(fig. 15).**

Adressage des micro-instructions

Pour minimiser le temps d'exécution du microprogramme, on utilise : **l'adressage explicite.**

La méthode consiste à diviser la partie de la micro-instruction contenant l'adresse en deux champs. Le premier contient explicitement les poids forts de l'adresse, le second contient soit directement les poids faibles de l'adresse si cette dernière est connue, soit l'adresse d'un élément de registre qui les contient si elle est conditionnelle. C'est-à-dire, qu'au lieu de fonctionner séquentiellement, adresse mémoire par adresse mémoire + 1 comme le fait le compteur programme du microprocesseur, le système se branche à une adresse mémoire contenue dans la configuration binaire de la micro-instruction. Cette méthode est obligatoirement employée compte tenu du découpage par champs des micro-commandes.

Le deuxième champ de l'adresse est utilisé en général de la façon suivante :

— Il contient 2 ou 3 bits d'information et 1 bit d'indice.

— Si l'indice est à 0, les 2 ou 3 bits représentent les poids faibles de l'adresse.

— Si l'indice est à 1, les 2 ou 3 bits adressent un registre qui contient les poids faibles de l'adresse. Dans la majorité des systèmes microprogrammés, cet indice est le bit Z du registre d'état.

En conclusion, une micro-instruction de 2 octets par exemple, est divisée de la façon représentée en **figure 16.** ■

Jean FREMAUX *

** Jean FREMAUX appartient à la direction générale de l'Ecole Centrale d'Electronique et assure les cours « microprocesseurs ».*

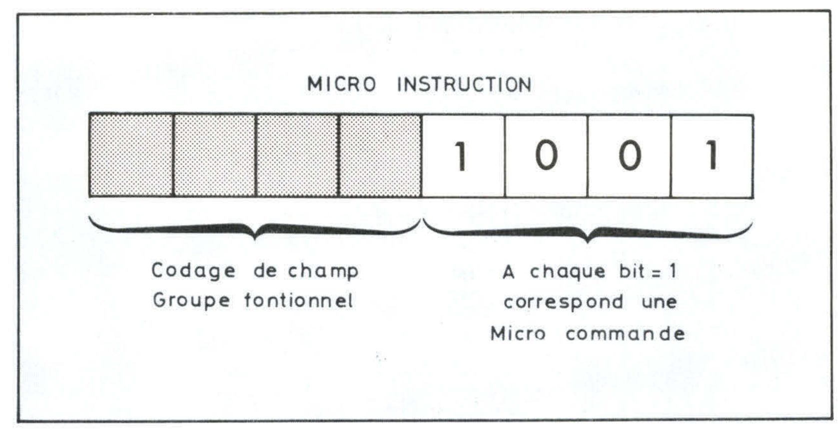
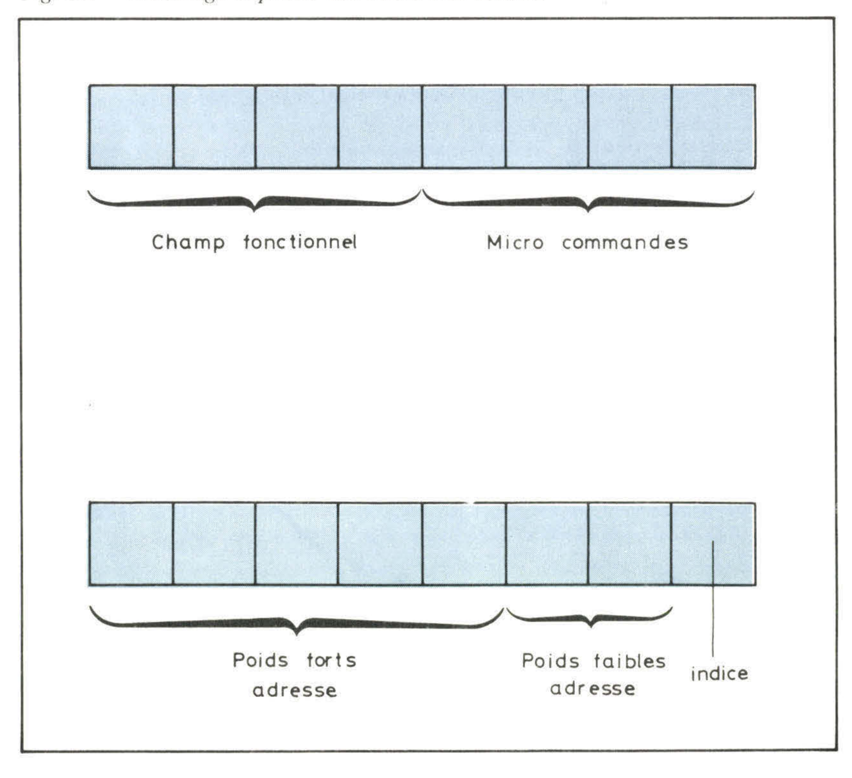


Fig. 15. – Les micro-instructions sont codées par champs. Un champ est un groupe de bits associé à une fonction. Dans un champ, chaque bit correspond à une micro-commande.

Fig. 16. – Adressage explicite des micro-instructions.



Les micro-ordinateurs dans la vie quotidienne

Le « Concours Micro »

Quand la science prend conscience d'elle-même

Dans « La fonction sociale de la Science aujourd'hui », introduction d'un volume dressant le bilan des recherches scientifiques les plus actuelles, Jean-Jacques Salomon s'interroge sur les rapports de la Science et de la Société.

En effet, alors que nous vivons dans un monde essentiellement déterminé par les progrès de la science et de la technologie, et bien qu'il nous soit difficile de renier tout l'acquis qu'a pu en tirer notre société moderne, il est intéressant aujourd'hui de constater qu'à cet égard le scepticisme l'emporte plus souvent que l'optimisme.

Au siècle dernier, l'activité scientifique et technique était considérée comme un facteur décisif du progrès humain.

A présent, cet optimisme est remis en cause : le développement scientifique et technologique n'engendre pas nécessairement le progrès lorsque l'on doit en son nom payer pour tribu la dislocation de notre société.

De ce fait, on en vient à l'idée que la science ne devra plus être déterminante de notre société, mais être déterminée dans ses orientations et ses applications par l'obtention d'un consensus propre à traduire les aspirations des individus et non celles de quelques experts.

La conclusion à cette introduction appartient à J.D. Bernal pour qui « La révolution scientifique est entrée dans une nouvelle phase : elle a pris conscience d'elle-même ».

L'informatisation de la société

L'apparition d'une nouvelle technique telle que l'informatique a nécessairement des incidences économiques et sociales qui amènent la société à se prononcer sur des choix à la fois politiques et idéologiques : il importe que notre société soit capable d'en promouvoir et d'en maîtriser les applications afin de toujours tendre vers le mieux-être de tous.

Rechercher les moyens pour conduire l' « Informatisation de la société », telle est la préoccupation du rapport Nora. L'objet de notre propos n'est pas d'entrer dans une analyse de ce document, mais ce dernier existe simplement pour nous montrer que nous sommes dès à présent engagés dans ce processus de réflexion.

Revenons à ces incidences économiques et sociales de l'informatique : Quelles peuvent-elles être ?

Sur le plan économique :

Le volume des données créées par l'homme a connu une augmentation rapide, ne serait-ce qu'en raison de la croissance démographique. Pour en assurer le traitement, l'informatique apporte un outil efficace dont l'utilisation accélère la mutation de notre société. Ainsi, on assiste à une forte diminution de la main-d'œuvre dans les secteurs **primaires *** et **secondaires ***, mais aussi à une croissance des services et des activités dont la matière première essentielle est constituée par l'information.

Dans ce dernier secteur, que l'on peut qualifier « d'informationnel » et pour lequel 45 % de la population active française travaille déjà, il apparaît évident que l'informatique constitue l'unique moyen propre à en augmenter la productivité sur qui, nous avons jusqu'à maintenant fondé notre progrès économique.

Ceci veut dire que nous nous trouvons devant une nécessité économique et que plutôt que de l'éviter, il est temps de rechercher comment il est possible qu'elle devienne une source de bienfaits pour l'individu.

Sur le plan social :

A la **lueur** de ce qui précède, l'informatique doit nous faire passer d'une société industrielle à une société d'information à haute productivité où les conflits se porteront de préférence sur les facteurs culturels.

L'apparition de l'écriture sous les **Sumériens *** allait apporter une mutation décisive de l'humanité sans que pour autant ceux-ci en prennent conscience.

Aujourd'hui, l'informatique est peut-être à l'origine d'un phénomène comparable : extension de la mémoire, prolifération et mutation des systèmes d'information.

Le vocabulaire d'accès à la machine se banalisant, l'informatique induira des effets sur le langage et la syntaxe.

Où s'arrêtera la communication informatisée maintenant que les ménages commencent à pouvoir s'équiper en micro-ordinateurs ?

Que deviendra l'écriture traditionnelle ?

Notre culture individuelle faite principalement de l'accumulation de connaissances ponctuelles va se trouver bouleversée, désormais, plus que par son habileté à chercher et à utiliser ces informations. Les concepts l'emporteront sur les faits, les itérations sur les récitations.

Pour autant que ces considérations ne soient pas l'objet d'espérances utopiques ou de craintes inhibitrices, il n'en demeure pas moins qu'elles sont révélatrices d'un problème posé à notre société par la technologie.

De plus, il y a de fortes chances pour que, dans l'état actuel de nos connaissances, chacun ne soit pas encore en mesure de pouvoir apprécier vers quelles orientations il serait bon de guider les applications de cette technique et suivant quelles modalités.

Jusqu'à maintenant, l'informatique a toujours été coupée du grand public. Elle est perçue de la façon la plus défavorable qui soit.

Mais pour se prononcer en toute sérénité, il importe avant toute chose que l'opinion publique, de qui devra se dégager un consensus, dispose d'une information et d'une formation suffisante.

Peut-on répondre à ce besoin élémentaire ?

— OUI !

La micro-informatique ou l'ordinateur-gadget ?

Le progrès technologique vient de mettre à présent à notre disposition une informatique de dimension humaine, dite micro-informatique.

Grâce à une extrême miniaturisation intervenue dans l'informatique traditionnelle, suivie d'une diminution des prix tout aussi spectaculaire, l'ordinateur à usage personnel devient réalité. A titre d'exemple, un ordinateur qui, il y a quinze ans se situait au sommet de la technologie, occupait une **pièce entière.**

Aujourd'hui, un microprocesseur de quelques millimètres de côté renferme la même puissance de calcul. A performances identiques, un composant qui coûtait 350 francs il y a dix ans vaut maintenant un centime.

Cette mutation rend possible à présent la fabrication de petits ordinateurs puissants et bon marché, désormais accessibles à une multitude d'utilisateurs, qui de ce fait conduiront à une rapide expansion de l'informatique. Loin de demeurer le privilège des grands organismes, il s'ouvre à des centaines de milliers d'usagers, petites entreprises, professions libérales, foyers, etc., un champ d'applications innombrables, à peine exploré.

Peut-être que si nous pouvons dire que l'informatique marque un tournant dans l'histoire de l'humanité au même titre que l'écriture, il nous sera permis d'aller plus loin dans cette analogie en ajoutant que la micro-informatique est à l'informatique ce que l'imprimerie de Gutenberg fut à l'écriture.

L'informatique se démocratise enfin.

Par son caractère personnel et personnalisable, il revient à chacun de l'utiliser librement, pour ses propres besoins et ses propres satisfactions.

Ramenée à une grande simplicité d'utilisation, cette informatique en cours de domestication a le grand mérite d'être accessible aux non-informaticiens ; finis les langages hermétiques, puisque le langage « Basic » s'apprend en quelques jours.

Il paraît maintenant que par un effort de sensibilisation et d'initiation à l'utilisation de micro-systèmes, nous construisons un pont qui permettra de franchir le fossé qui pendant trop longtemps a séparé le grand public de l'informatique traditionnelle.

Toutefois, on peut se demander si la construction de ce pont repose sur des assises solides.

N'y a-t-il pas quelques écueils qui pourraient compromettre la réalisation de ce rêve un instant esquissé ?

Ce que redoute la « Mission à l'informatique » c'est que rapidement nous nous engagions dans une « gadgétisation » comme cela s'est fait pour les calculateurs de poche.

Alors que ces instruments ont permis à certains de se surpasser dans le traitement des problèmes complexes, ce qui est un réel progrès, on trouve des utilisateurs qui n'ont pas fourni l'effort intellectuel minimum, nécessaire pour tirer parti des possibilités de cet outil. Ces personnes, pour qui toute opération arithmétique simple passe nécessairement par l'utilisation de leur calculateur, en viennent à régresser dans la connaissance et la maîtrise de l'arithmétique.

Que doit-on faire ?

Il est évident qu'il faut à tout prix éviter de sombrer dans la facilité.

Le comportement de l'individu devant son micro-ordinateur ne doit pas être passif.

« Le Concours Micro » tremplin de la micro-informatique

C'est la raison pour laquelle la « Mission à l'Informatique » a décidé de créer le « Concours Micro » afin d'inciter et d'aider tous ceux que la micro-informatique attire, à trouver les moyens d'une action résolument active.

• La « Mission à l'Informatique » offrira des prix pouvant atteindre 50 000 F et qui vous permettront d'acheter votre matériel tant souhaité. Des aides financières et techniques pourront vous être accordées pour perfectionner ce que vous avez déjà entrepris.

Un prix spécial de 30 000 F récompensera la meilleure étude consacrée aux économies d'énergie.

• Pour concourir, il vous suffit d'inventer et de réaliser des utilisations de l'informatique personnelle.

Pour cela, deux possibilités :

• Etablir un projet

ou • Réaliser une application

et les soumettre au jury du concours.

On entend par **projet** une idée d'utilisation d'un micro-ordinateur dans la vie quotidienne.

Ce projet prendra la forme d'une description détaillée comprenant les organigrammes du logiciel et une évaluation complète du matériel nécessaire.

On entend par **réalisation** l'utilisation effective d'un micro-ordinateur dans la vie quotidienne. Elle doit comprendre un logiciel s'appuyant sur un matériel.

La catégorie réalisation comprend également des développements ou modifications de systèmes déjà existants.

• Afin de faciliter l'information des concurrents et de leur permettre de tester leurs projets ou réalisations, la « Mission à l'Informatique » a choisi des « Points d'essai concours Micro ». Ils vous fourniront des informations techniques sur le matériel qu'ils possèdent ou qu'ils diffusent et seront en mesure de vous dispenser de nombreux conseils.

Enfin, chose importante, vous trouverez chez eux, un micro-ordinateur mis gracieusement à votre disposition pour vos essais.

Ainsi, fidèle à son éthique, la « Mission à l'Informatique », à qui revient l'heureuse initiative de ce concours, va par la mise en place de ce vaste réseau national de « Points d'essais » contribuer à une large décentralisation et démystification de l'informatique.

A cela, il convient d'ajouter, et c'est là l'un des grands mérites de ce concours, que par le puissant appel à la créativité dont il procède, des milliers de personnes de tout âge vont se trouver engagés dans une compétition qui ne manquera pas, à terme, d'asservir l'informatique aux besoins de notre société.

De ce seul fait, la micro-informatique ne peut plus sombrer dans un marché factice. Grâce à l'élan qu'aura suscité cette action des pouvoirs publics, les utilisateurs constitueront un bloc éclairé dont l'exigence et la pertinence des demandes conduiront à une constante amélioration des systèmes. ■

J.-J. WANÈGUE

** Le secteur primaire regroupe la production agricole et la production de matières premières.*
** Le secteur secondaire comprend, d'une part la production d'énergie et, d'autre part, les industries de transformation ; à cela, il faut ajouter l'automobile, le bâtiment et les industries chimiques et textiles.*
** Sumériens : peuple qui s'établit dans la basse vallée de l'Euphrate au IVe millénaire av. J.-C.*

Bibliographie

— La fonction sociale de la science aujourd'hui de Jean-Jacques SALOMON Encyclopoedia Universalis

— L'informatisation de la Société de Simon NORA et Alain MINC Documentation Française

— Ordinateurs et Options sociales Ouvrage collectif Collection Impact Science et Société : UNESCO

La mission à l'Informatique

La « Mission à l'Informatique » fait partie du Ministère de l'Industrie.

Elle a pour objectif de préparer et mettre en œuvre une politique d'utilisation de l'informatique.

Plus précisément, elle a pour attributions :

- d'aider les utilisateurs en leur founissant les moyens leur permettant une plus grande maîtrise de l'informatique ;
- d'évaluer l'impact de l'informatique sur son environnement économique et social ;
- de participer à la définition et à la mise en œuvre d'une politique de formation et d'enseignement dans le domaine de l'informatique.

La Mission à l'Informatique dirigée par J.-Claude Pelissolo comprend 18 chargés de mission dont les travaux couvrent un très vaste domaine allant de l'informatisation des Administrations à la Formation à l'informatique.

« Mission à l'Informatique », 24, rue de l'Université, 75007 Paris.

Ce que peut vous apporter « Micro-Systèmes »

Si vous êtes seul et que vous souhaitez vous associer avec d'autres concurrents.

Si vous ignorez où il vous serait possible de tester vos réalisations.

Si votre projet vous pose encore de nombreux problèmes.

Et qu'en réponse à tout ceci vous aimeriez recevoir les conseils des spécialistes de Micro-Systèmes.

Faites-nous connaître vos questions en nous écrivant à :

Micro-Systèmes
Concours « Micro »
15, rue de la Paix, 75002 Paris

N'oubliez pas de nous indiquer votre nom, votre adresse, votre profession, votre âge.

Si vous êtes étudiant, faites-nous connaître le nom de votre établissement d'enseignement.

Enfin, si vous cherchez à vous associer avec d'autres concurrents, il est important de savoir vers quel domaine vous envisagez d'orienter votre recherche.

Si vous désirez recevoir le dossier d'information sur le concours, renvoyez le bulletin ci-dessous à :

Concours « micro »
Mission à l'Informatique
24, rue de l'Université, 75007 PARIS

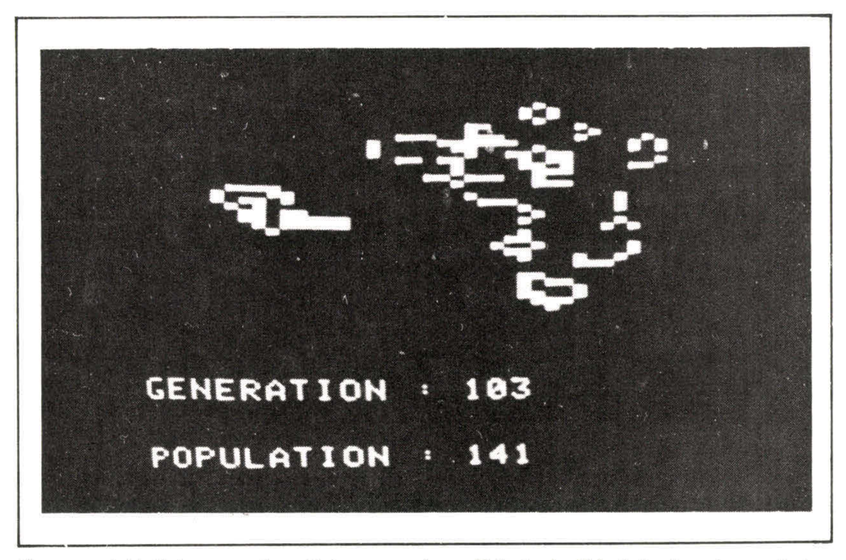
Nom ....
Adresse ....
Age (1) .... Profession (1) ....
(1) Facultatif.

Les vœux de « MICRO-SYSTÈMES » vous accompagnent.

Jeu de la vie

Il y a une trentaine d'années, John Von Neumann, l'un des plus éminents artisans de la réalisation des ordinateurs, se posa une question fondamentale : une machine peut-elle se reproduire ?

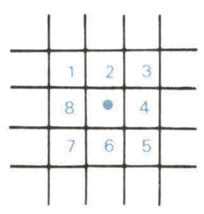
La réponse est oui. Mais Von Neumann ne répondit pas en exhibant une machine réelle. Il élabora sur le papier une machine abstraite : un automate cellulaire. C'est un damier où des pions vivent et meurent selon des règles logiques, en fonction de leur voisinage. Par la suite cette théorie des « automates cellulaires » s'est développée, et plusieurs mathématiciens ont exploré les lois pouvant être imposées aux pions du damier. Parmi celles-ci, John Horton Conway crée une série de lois qu'il baptise lui-même : *La Vie.*



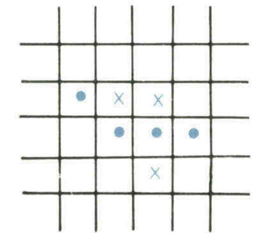
Vous avez initialisé un type de société sur votre écran T.V. Après 103 générations, la population comporte 141 individus et présente la configuration ci-dessus.

La Vie évolue sur un damier plan illimité. Elle organise les morts et les naissances de pions sur les cases. Un pion naît sur une case lorsque les conditions sont favorables. Il meurt lorsque les conditions sont devenues intolérables pour lui. Plus précisément, les lois sont les suivantes :

— Chaque pion possède 8 voisins.



— Naissances : il naît un pion dans toute case vide en contact avec exactement 3 cases occupées. Ainsi pour cette société de 4 pions, il y a 3 naissances indiquées par des X.

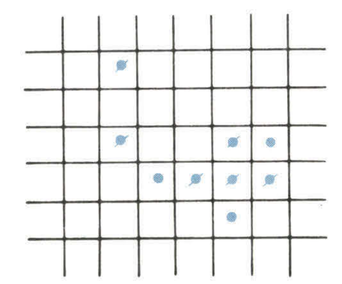


— Morts : un pion peut mourir :

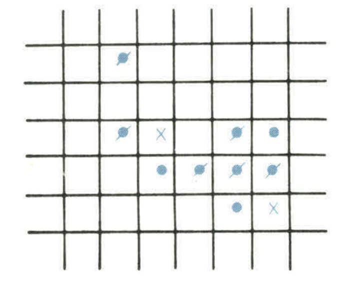
● Soit par isolement s'il est en contact avec un seul autre pion ou aucun ;

● Soit par étouffement s'il est en contact avec quatre pions ou plus.

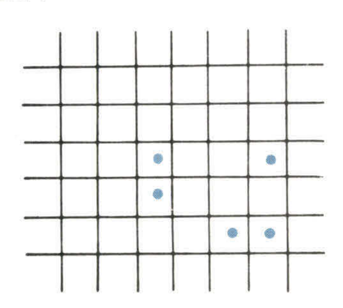
Ainsi dans la société suivante, deux pions meurent par isolement et quatre par étouffement (les morts sont barrés).



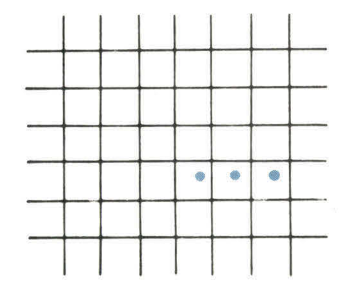
Dernière règle, les sociétés évoluent par sauts successifs, appelés générations. A chaque nouvelle génération, les morts et les naissances se produisent simultanément, sans influer les unes sur les autres. Ainsi la mort d'un pion ne l'empêche pas de participer à la naissance d'un autre. Dans la société précédente, il se produit 2 naissances et 6 morts :



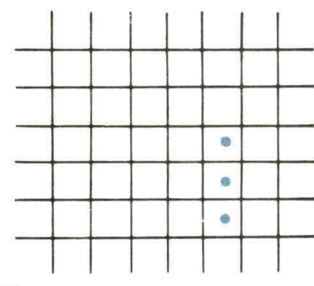
La génération suivante est donc :



Ce qui donne 5 morts et 3 naissances, et la troisième génération devient :

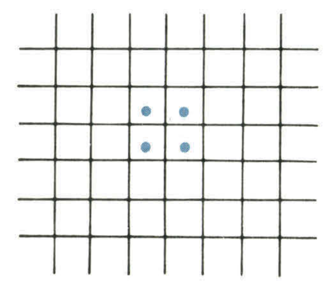


Génération suivante :

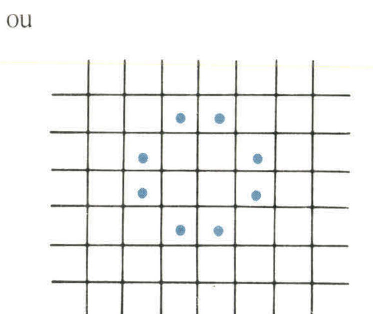


Etc...

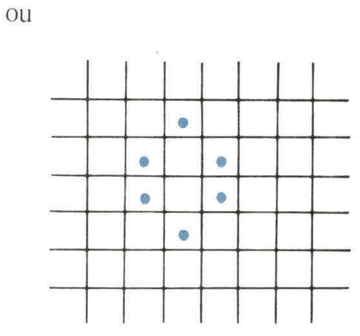
La Vie connaît un grand succès depuis sa parution en 1970 dans Scientific American. De nombreux chercheurs amateurs étudient des sociétés particulières soit avec une feuille de papier et un crayon, soit sur ordinateur (comme au Palais de la Découverte). Certains types de société présentent des caractéristiques singulières, comme celles qui restent stables :



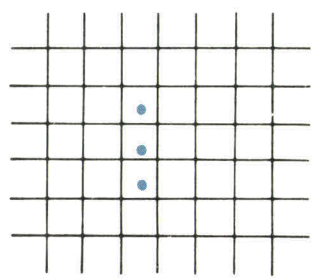
ou

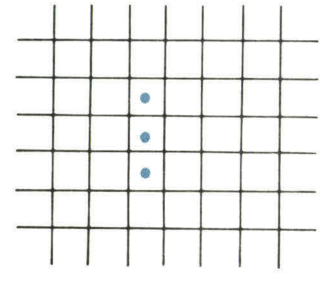


ou



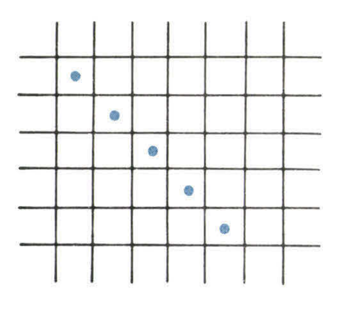
D'autres oscillent indéfiniment avec une période plus ou moins longue. L'exemple le plus simple en est le « clignotant ».

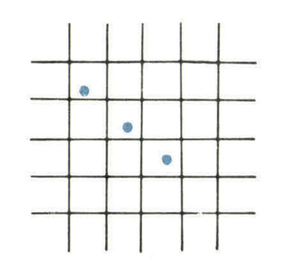


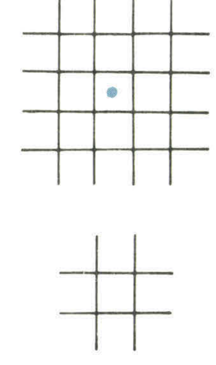


Etc...

D'autres meurent lentement comme la « fusée » :



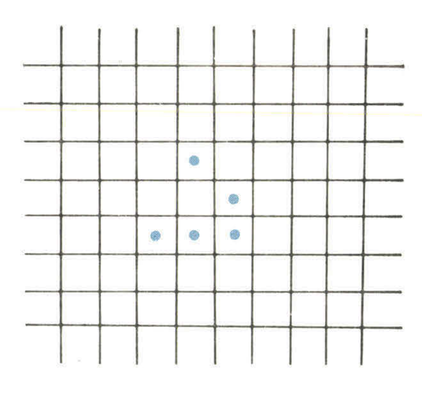




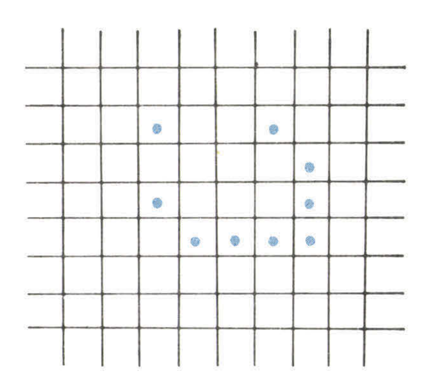
Mort

D'autres populations peuvent se déplacer :

— Soit diagonalement comme celle-ci.



— Soit horizontalement comme celle-ci :



Plusieurs questions sont alors posées :

— Existe-t-il une population qui puisse s'étendre indéfiniment ?
— Que se passe-t-il en cas de choc entre une population stable et une qui se déplace ? Ou même entre deux populations qui se déplacent ?

A vous de faire l'essai à l'aide du listing en Basic ci-dessous.

Le programme lui-même occupe 2 K octets en mémoire, mais le terrain sur lequel évolue **La Vie** peut demander beaucoup plus de place que cela. Les dimensions fixées à la ligne 110 peuvent donc être modifiées en fonction de la taille mémoire et aussi de l'écran servant à visualiser ces évolutions. Les autres lignes du programme restent inchangées.

Attention au temps d'exécution. Celui-ci peut devenir important si La Vie se déroule sur un grand terrain.

Mais regardons d'un peu plus près le listing : Les lignes 200 à 340 servent à rentrer au clavier le dessin correspondant à la première génération. Pour qu'une réponse soit prise en compte correctement par le Basic, il faut qu'elle commence par un caractère visible. D'où l'obligation des parenthèses, qui sont ensuite éliminées à la ligne 300.

Le calcul de la génération suivante se fait de la ligne 1 000 à la ligne 1 410 en comptant les voisins de chacune des cases. Il faut éviter dans ce cumul les voisins qui se trouvent à l'extérieur du tableau réservé, d'où les nombreux tests de limites sur I (abscisse) et J (ordonnée). La création d'un nouvel individu dépend du nombre de voisins N et de la présence éventuelle d'un pion à la génération précédente : ligne 1 330.

Il ne reste plus qu'à afficher le résultat (lignes 1 440 à 1 530). Ceci de nouveau à l'aide de deux boucles imbriquées, l'une comptant les lignes et l'autre plaçant les pions à l'intérieur de chaque ligne.

Le jeu se termine lorsqu'il n'y a plus de survivants, soit parce que la population s'est déplacée hors de l'écran, soit parce que tous les individus sont morts. ■

H. EYMARD-DUVERNAY

LISTING DU PROGRAMME

```
100  REM JEU DE LA VIE
110  H = 30 : V = 15 : REM DIMENSIONS DU TERRAIN
120  DIM A (H, V), C (H, V)
130  PRINT
140  PRINT TAB (10); « JEU DE LA VIE »
150  PRINT
160  PRINT « DESSINEZ LA PREMIÈRE GÉNÉRATION »
170  PRINT « CHAQUE LIGNE DOIT ÊTRE ENTRE PARENTHÈSES »
180  PRINT « TAPEZ UN ESPACE S'IL N'Y A PAS DE PION »
190  PRINT « ET UN * POUR CHAQUE PION PRÉSENT »
200  PRINT
210  I = 0 : G = 1
220  I = I + 1
230  IF I > V THEN 1000
240  PRINT « LIGNE » ; I ;
245  INPUT A$
250  IF A$ = « » THEN 1000
260  A$ = LEFT$ (A$, H + 2)
270  FOR J = 1 TO H
280  A (J, I) = 0
290  NEXT J
300  FOR J =2 TO (LEN (A$) - 1)
310  IF MID$ (A$, J, 1) < > « * » THEN 330
320  A (J - 1, I) = 1
330  NEXT J
340  GOTO 220
990  REM CALCUL DE LA POPULATION SUIVANTE
1000 FOR I = 1 TO H
1010 FOR J = 1 TO V
1020 C (I, J) = 0
1030 NEXT J
1040 NEXT I
1050 G = G + 1 : P = 0
1060 FOR I = 1 TO H
1070 FOR J = 1 TO V
1080 N = 0 : REM NOMBRE DE VOISINS
1090 IF J = 1 THEN 1 130
1100 IF A (I, J - 1) = 1 THEN N = N + 1
1110 IF I = H THEN 1 170
1120 IF A (I + 1, J - 1) = 1 THEN N = N + 1
1130 IF I = H THEN 1 170
1140 IF A (I + 1, J) = 1 THEN N = N + 1
1150 IF J = V THEN 1 210
1160 IF A (I + 1, J + 1) = 1 THEN  N = N + 1
1170 IF J = V THEN 1 210
1180 IF A (I, J + 1) = 1 THEN N = N + 1
1190 IF I = 1 THEN 1 300
1200 IF A (I - 1, J + 1) = 1 THEN N = N + 1
1210 IF I = 1 THEN 1 300
1220 IF A (I - 1, J) = 1 THEN N = N + 1
1230 IF J = 1 THEN 1 300
1240 IF A (I - 1, J - 1) = 1 THEN N = N + 1
1290 REM ELIMINATION DES CAS ININTÉRESSANTS
1300 IF N < 2 THEN 1 350
1310 IF N > 3 THEN 1 350
1320 IF N = 3 THEN 1 340
1330 IF A(I, J) = 0 THEN 1 350
1340 C (I, J) = 1 : P = P + 1
1350 NEXT J
1360 NEXT I
1370 FOR I = 1 TO H
1380 FOR J = 1 TO V
1390 A (I, J) = C (I, J)
1400 NEXT J
1410 NEXT I
1420 IF P = 0 THEN 1 600
1430 PRINT : REM DESSIN DE LA NOUVELLE GÉNÉRATION
1440 FOR I = 1 TO V
1450 FOR J = 1 TO H
1460 IF A (J, I) = 1 THEN PRINT « * » ;
1470 IF A (J, I) = 0 THEN PRINT «   » ;
1480 NEXT J
1490 PRINT
1500 NEXT I
1510 PRINT
1520 PRINT « GENERATION » ; G ; « POPULATION = » ; P
1530 PRINT
1540 GOTO 1000
1590 REM FIN DE PARTIE
1600 PRINT « IL N'YA PLUS AUCUN INDIVIDU »
1610 PRINT « A LA GÉNÉRATION SUIVANTE »
1620 PRINT
1630 INPUT « VOULEZ-VOUS RECOMMENCER », A$
1640 IF LEFT $ (A$, 1) = « 0 » THEN 200
1650 IF LEFT$ (A$, 1) < > « N » THEN 1 630
1660 PRINT
1670 PRINT TAB (10); « AU REVOIR !! »
1680 PRINT
1700 END
```

Séminaires

Formation microprocesseur

La société I.C.S. (Integrated Computer Systems) organise des stages de formation les 6, 7, 8, 9 février et les 3, 4, 5 et 6 avril à Paris.

Trois sujets seront traités au cours de ces journées :

— **Organisation de projets à microprocesseur :** De la conception à la réalisation, et de l'assurance qualité à la maintenance (1 journée).

— **Microprocesseurs et micro-ordinateurs :** Introduction technique globale (1 journée).

— **Cours pratiques sur micro-ordinateur :** Programmation et interfaces (2 journées).

Pour de plus amples informations :

ICS France Sarl
90, av. Albert-1er, 92500 Rueil-Malmaison
Tél. : (01) 749.40.37.

Le mouvement Jeunes-Science et l'Institut Supérieur d'Electronique du Nord organisent à **Dunkerque** un séminaire d'initiation à la micro-informatique au cours des vacances de Pâques 1979 :

— du lundi 16 avril à 14 heures au samedi 21 avril à 12 heures.

Le nombre de participants est limité à 50. L'âge minimum est de 17 ans. Les frais d'inscription au séminaire sont de 350 F par personne pour la semaine. Ils couvrent l'hébergement collectif, la documentation et la totalité des repas du lundi soir au samedi matin.

Les inscriptions sont ouvertes dès maintenant. Il est vivement conseillé de s'inscrire avant la fin janvier, étant donné le nombre limité de participants.

Pour s'inscrire, envoyer :
— nom, prénom, âge,
— enveloppe timbrée pour la réponse,
— et bien spécifier à quel atelier vous souhaitez participer du jeudi au samedi.

Notre adresse :

Séminaire micro-informatique
Jeunes-Science Dunkerque
B.P. 1-501, 59383 Dunkerque cedex.

Octets

Octets a pour vocation première de réunir le plus grand nombre possible d'amateurs de micro-informatique, quels que soient les horizons dont ils viennent, et qu'ils possèdent ou non des machines. Chacun pourra y communiquer avec les autres, dans le cadre de réunions hebdomadaires ou certains essaieront de faire partager leur passion, alors que d'autres découvriront progressivement les merveilles d'une technique qui leur est nouvelle. De plus, un courrier bimestriel nous permettra de resserrer les liens qui nous uniront avec les adhérents de province, de leur faire bénéficier des tours de main découverts au cours de nos réunions, et de faire le point sur les nouveaux matériels disponibles sur le marché.

Nous serons, par ailleurs, très heureux de lier des contacts étroits avec toutes associations menant des activités identiques à la nôtre.

Marc Hervais : 329-08-05.
Jacques Lipka : 622.35.11.

L'Ecole Supérieure d'Ingénieurs de Marseille organise des sessions de formation continue dans le domaine de l'Electronique et de l'Informatique en 1979 :

— Conception des circuits logiques combinatoires et séquentiels par les méthodes de l'analyse binaire, du 12 au 16 février.

— Initiation aux microprocesseurs et à leur programmation, du 15 au 19 janvier.

— Réalisation d'automates en logique câblée et microprocesseurs, du 28 mai au 1er juin.

— Initiation à l'informatique industrielle. Tous les mardis du 20 février au 22 mai.

— Programmation structurée, du 23 au 27 avril.

Obtention des programmes détaillés et inscriptions :

Service des Relations Industrielles
Ecole Supérieure d'Ingénieurs de Marseille
110, bd de la Libération, 13004 Marseille
Tél. : (91) 47.07.96.

Cours de formation Intersil

Intersil organise périodiquement des cours de formation sur le microprocesseur IM6100, avec manipulation sur système Intercept. Par ailleurs Intersil peut organiser des cours spéciaux de 2 ou 3 jours sur demande.

Ces séminaires ont une durée de trois jours. Le premier jour est optionnel et est destiné à l'initiation aux microprocesseurs afin que les ingénieurs ou techniciens non spécialisés suivent avec profit les deux jours suivants.

Calendrier des cours 1978/1979 : 27, 28 février et 1er mars 1979 ; 5, 6 et 7 juin 1979.

Lieu : Trianon Palace-Hôtel, 1, boulevard de la Reine. 78000 Versailles.

Prix : 2 jours (cours 6100) : 1 000 F HT ; 3 jours (introduction plus cours 6100) : 1 500 F HT.

Intersil.
3, rue de Marly - 78000 Versailles - Tél. : (1) 953.47.08.

Bibliographie

Le Hardsoft



Le Hardsoft ou la « Pratique des microprocesseurs » de Ouaknine et Poussin édité chez ETSF est un ouvrage d'initiation et de formation particulièrement destiné aux électro-

niciens et informaticiens non spécialistes.

Cette 2e édition revue et augmentée aborde les principes généraux, le fonctionnement et le jeu d'instruction d'un système construit autour du 8080 A.

Les techniques de programmation sont introduites par de nombreux exemples et applications réelles avec schémas et programmes.

Les auteurs traitent ensuite des micro-ordinateurs en un boîtier de la famille 8048, du microprocesseur Z80 et d'un microprocesseur 16 bits le 8086.

Un volume broché de 254 pages, format 15 x 21, 98 schémas et tableaux, couverture couleur. Prix : 65 F.

En vente chez votre libraire habituel ou à la Librairie Parisienne de la Radio : 43, rue de Dunkerque 75010 Paris.

Théorie des programmes schémas, preuves, sémantique par C. Livercy

La construction des programmes s'accommode de moins en moins d'empirisme et d'imprécision, et il devient vital pour l'analyste et le programmeur d'avoir un certain recul par rapport à cette technique qu'est la programmation. Cet ouvrage, le premier en langue française traitant de ce sujet, répond à ce souci en donnant une présentation synthétique et approfondie des principaux aspects de la théorie des programmes. Des théories développées il y a déjà une douzaine d'années, telle que celle des schémas de programmes, y côtoient des domaines encore en plein développement comme celui de la sémantique des langages de programmation ou la théorie du point fixe.

Le choix des concepts introduits, la présentation claire mais volontairement schématique des démonstrations, les allusions fréquentes aux applications et la présence de nombreux exemples et exercices corrigés font de ce livre un instrument de travail et de culture, aussi bien pour les enseignants et étudiants en informatique que pour tous les praticiens de l'informatique désireux de réfléchir sur leur activité.

Collection « Dunod Informatique », phase formation, 15,5 x 24, 340 pages. Broché : 110 FF.

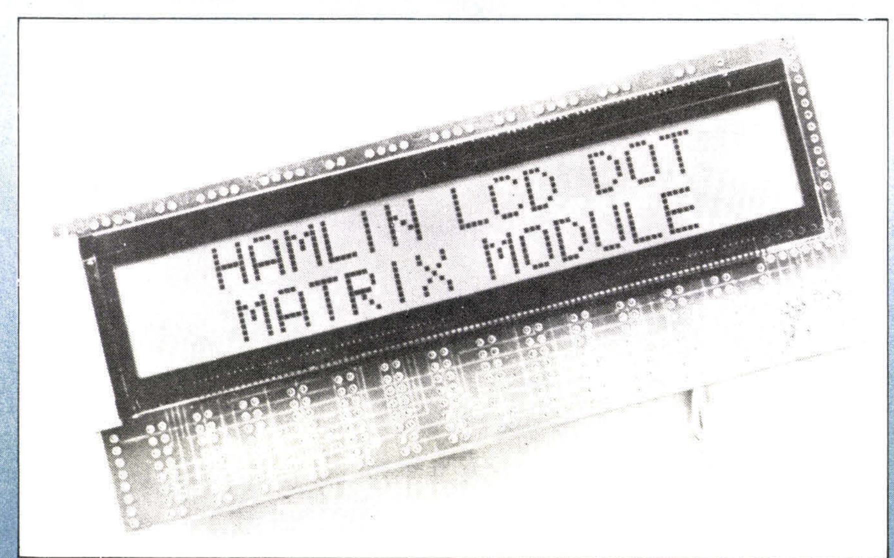
Introduction à A.P.L. par S. Pommier

APL est un langage de programmation qui se distingue par sa concision et la puissance de ses opérateurs. Son apprentissage et sa pratique ne requièrent aucune connaissance préalable en informatique, car il a été conçu de façon à ce que l'utilisateur puisse tout ignorer de la structure et du fonctionnement de l'ordinateur avec lequel il dialogue.

Cet ouvrage constitue une introduction à APL. Fruit d'une expérience pédagogique de plusieurs années, il est destiné à tous ceux qui désirent se familiariser avec ce nouveau langage, en saisir les originalités et la philosophie. Rédigé avec un grand souci de clarté, il permet une première découverte du monde de l'informatique. De nombreux exercices, accompagnés de leurs solutions, sont proposés pour tester les connaissances progressivement acquises au fil des chapitres.

DUNOD 1978.
Collection « Dunod Informatique » - phase formation.
15,5 x 24, 152 pages. Broché. 46F.

Hamlin lance sur le marché une matrice point par point



Cette matrice, composée de deux lignes ayant chacune 16 caractères, est destinée à équiper les terminaux de point de vente, les visualisations de micro-ordinateurs ou toutes applications identiques.

Le module de référence 3200 comporte 16 caractères de 5 x 7 points chacun d'une part, et un circuit de commande multiplexé monté sur un circuit imprimé, d'autre part.

Les dimensions du module sont :
162 x 67 x 15 mm

celles des caractères :
8,6 x 6 mm.

Un circuit de compensation réduit les interférences entre affichages permettant un degré de contraste meilleur que celui généralement obtenu par les autres méthodes de multiplexage.

Le module étant compensé, au niveau du circuit de commande, en température, peut être utilisé entre 0 et 50 °C.

La tension d'alimentation est de 12 V CC pour un courant de 12 mA.

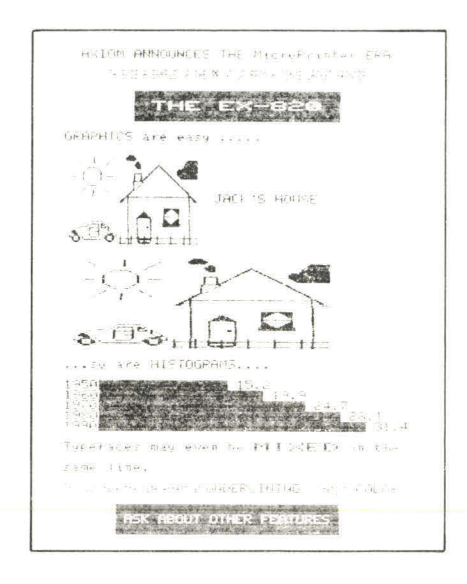
Le module 3200 point par point Hamlin se monte sur panneaux. Les bords de l'ouverture étant échancrés de façon à obtenir un meilleur contraste.

Dans ces conditions, l'angle de vue est de 10 et 40° de face et de ± 45° de côté. Le contraste est de 20 : 1.

Hamlin Electronics
3, Sente des Dorées, 75019 Paris
Tél. : 202.53.17.

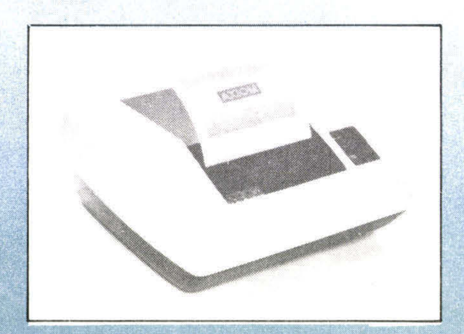
Auctel représente les Imprimantes Olivetti NIP 18

Auctel vient de prendre la représentation exclusive pour la France des blocs Imprimantes Olivetti sur papier électrosensible NIP 18.



Caractéristiques :
- format variable jusqu'à 24 c/ligne,
- faibles dimensions 175 x 45 x 80 mm,
- vitesse 2 lignes/seconde,
- papier largeur 60 mm.

Trois modèles sont commercialisés :
- Le mécanisme seul, P.U. HT 515 F.
- Le mécanisme avec circuit d'écriture et de commande moteur, P.U HT 1 470 F.
- Le mécanisme avec circuit d'écriture de commande moteur piloté par microprocesseur, P.U. HT 1 950 F.



Auctel pense vendre la première année 5 000 unités sur les marchés OEM.

Auctel
143, rue des Meuniers, 92220 Bagneux
Tél. : 664.10.50+.

Micro-ordinateur RTC

Dans le cadre des accords conclus en milieu de 1977 entre les sociétés Signetics et Intel concernant la famille micro-ordinateur 8048, et à la suite de l'extension de ces accords RTC introduira sur le marché français, à la fin de cette année, le micro-ordinateur 8021.

Destiné aux applications de contrôle à grande diffusion telles que celles que l'on rencontre dans les domaines de l'électroménager, de l'automobile, des instruments de tests et de mesures, et des télécommunications, le 8021 est une version économique dérivée du 8048.

Tout comme les micro-ordinateurs 8048/8035 déjà commercialisés par RTC, le 8021 comprend une unité centrale 8 bits, 64 octets de mémoire RAM et 1024 octets de mémoire ROM.

Il ne comporte par contre que 21 lignes d'entrées/sorties, ce qui permet de le présenter dans un boîtier 28 broches plus économique.

Un compteur/temporisateur programmable ainsi qu'un système d'horloge et un oscillateur sont intégrés sur le cristal.

Le jeu de 60 instructions est dérivé de celui du 8048 et la tension d'alimentation peut varier de 4,5 V à 6 V.

R.T.C.
130, avenue Ledru-Rollin, 75540 Paris Cedex 11 - Tél. (1) 355-44-99.

Potentiomètre à commande numérique

Analog Devices annonce la commercialisation d'un potentiomètre à commande numérique BCD, le AD 7525. C'est en fait un convertisseur numérique/analogique à entrée 3 1/2 digits BCD. Ce convertisseur monolithique en technologie CMOS est en outre, multiplieur 2 cadrans, le signal contrôlé digitalement pouvant être positif, négatif, ou alternatif. L'application la plus évidente de ce AD 7525 est le positionnement d'un

niveau analogique à partir de roues codeuses montées en façade d'un appareil.

Le AD 7525 permet une résolution de 0,1 %, une dynamique de sortie de 0,000 à 1,999 fois la tension d'entrée avec une précision absolue de 0,02 %.

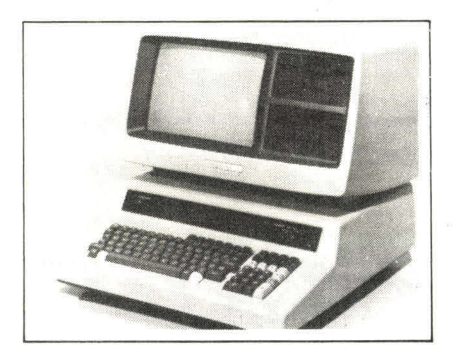
Présentation boîtier DIL 18 broches plastique ou céramique. Gamme de température : civile, industrielle et militaire.

Analog Devices
Centre d'Affaires Silic
12, rue Le Corbusier, bât. « Iéna », 94533 Rungis Cidex L 204
Tél. : 686.77.60
Télex 200156 A.N.A.

La gamme des micro-ordinateurs japonais SORD s'élargit

Le M203 MARK II

Conçu autour du Z80, ce micro-ordinateur, d'aspect compact, est un produit non extensible prêt à être mis en service ; il se compose de : 64 K



de mémoire RAM – 1 ou 2 disquettes de 350 K octets chacune – un écran de 24 lignes de 80 caractères – un interface entrées/sorties parallèles – 2 interfaces V 24 pour la connexion d'une imprimante et d'un modem – un clavier avec bloc numérique séparé et touches fonctions – une horloge interne.

En option, un compilateur BASIC, un macroassembleur et un FORTRAN IV sont disponibles.

Le M223 MARKII

Le M223 est un micro-ordinateur plus sophistiqué et extensible. Outre le composants du M203, ce modèle dispose d'un bus S100 sur lequel il est possible de connecter différents coupleurs (pour le contrôle de processus par exemple). Il possède également un interface disquette fonctionnant sur le canal d'accès direct à la mémoire (DMA) et peut supporter 4 disquettes de 350 K octets chacune.

Les options disponibles sont les suivantes : un convertisseur A/N et N/A – un coupleur graphique noir et blanc ou couleur – un coupleur pour traceurs de courbe, etc.

Interco : 734.73.35/566.67.36

Un nouveau convertisseur Numérique/Analogique

Le **AD565,** commercialisé par ANALOG / DEVICES est un convertisseur N/A monolithique de 12 bits rapide. Son temps d'établissement à 1/2 LSB est de 200 ns (temps de commutation : 30 ns). Sa référence interne de 10 V est réalisée par une zener « implantée » de haute stabilité. La dérive du gain ne dépasse pas 20 ppm/°C max. pour le modèle K (gamme t° = 0° à 70 °C) et 10 ppm/°C pour le modèle T (gamme t° = – 55° à + 125 °C).

Présentation du AD565 en boîtier DIL24 broches plastique ou céramique, compatible borne à borne avec le AD563 bien connu.

Index des Annonceurs

Le directeur de la publication J.P. Ventillard - Imprimerie La Haye-Mureaux, 78130 Les Mureaux. Tél. : 261-56-15
N° de Commission paritaire 61 025 - Dépôt légal n° 13 105

l'informatique à votre portée

avec les ordinateurs "HEATHKIT"

... deux systèmes complets, étudiés spécialement pour vous, techniciens, scientifiques, éducateurs, amateurs avertis, responsables d'entreprises.

- Système H8, mots de 8 bits, avec microprocesseur 8080 A et capacité de mémoire centrale extensible à 64 K mots (RAM statique).
- Système H11, mots de 16 bits, avec microprocesseur LSI 11, et capacité de mémoire centrale extensible à 64 K mots (RAM – MOS dynamique).
- Nombreux périphériques : Console de visualisation - Perforateur/lecteur de bande - Imprimante DEC LA 36 (30 cps) - Mémoire masse sur cassettes, disques souples - Interfaces I/0 séries, parallèles. Ce matériel est livré en KIT* ou en ordre de marche, avec logiciel très complet, comprenant même le Basic 8K (en option : Basic 12 K et gestion fichier) vous permettant de développer vos applications à l'infini.

* Les unités centrales (CPU) sont livrées câblées et testées par HEATHKIT.





KIT MICRO-ORDINATEUR d'initiation "6800"
Moniteur 1 K ROM - 256 x 8 RAM - Extension à 512 x 8 RAM - Affichage 6 digits, 7 segments - Clavier hexadécimal - Possibilité d'extension cassette, etc... - Livré avec alimentation stabilisée - Manuel en Anglais.

Prix : **1.990 F** (TTC) + port 20 F.

COURS sur les MICROPROCESSEURS
Le plus complet à ce jour, 8 chapitres, 800 pages, rédigé en Anglais, assorti d'exemples et expériences, à l'aide de 62 composants électroniques (y inclus).

Prix : **820 F** (TTC) + port 10 F.

Bon à découper, à adresser à :

FRANCE : Heathkit, 47 rue de la Colonie, 75013 PARIS, tél. 588.25.81
BELGIQUE : Heathkit, 16 Av. du Globe, 11.90 BRUXELLES, tél. 344.27.32

Je désire recevoir votre catalogue 16 pages couleurs, en Anglais, contenant tous renseignements sur : (cocher la case)

☐ Ordinateurs H 8 et H 11 ☐ Cours sur les microprocesseurs

Je joins 4 Francs en timbres, pour frais d'envoi

Nom, prénom ____________________

N° ________ Rue ____________________

Code postal ____________ Ville ____________

MS 01-79

Centres de démonstration :
PARIS (6e), 84 Bd Saint-Michel, tél. 326.18.91
LYON (3e), 204 rue Vendôme, tél. (78) 62.03.13